



INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

Ricardo Francisco Fernandes do Vale

APLICAÇÃO MULTI-DISPOSITIVO RESPONSIVA PARA O SETOR DA RESTAURAÇÃO

Mestrado em Engenharia de Software

Trabalho de Projeto efectuado sob a orientação do
Doutor António Miguel R. S. R. Cruz

Março de 2014

RESUMO

A ubiquidade e a diversidade de dispositivos disponíveis no mercado está a mudar a forma de como acedemos e partilhamos informação nos dias de hoje, sendo necessário oferecer experiências de utilização consistente para um crescente número de contextos.

Este relatório descreve o projeto Pratos em Cartaz, desenvolvido no âmbito do Projeto de Mestrado em Engenharia de Software. É uma plataforma para procurar pratos em destaque nos restaurantes perto da sua localização e para poder ajudá-lo a ir lá. É feita uma contextualização do projeto e apresentados alguns dados sobre a empresa envolvida no projeto. São também apresentadas algumas definições e conceitos e é feita uma análise do Estado da Arte sobre os principais *frameworks* responsivos atuais. É descrita a análise do problema do projeto, com apresentação dos seus requisitos, modelo de casos de uso e o modelo de domínio. O desenvolvimento do projeto é apresentado a partir da arquitetura da solução e sua interface assim como as tecnologias envolvidas. A terminar o relatório é feita a avaliação e discussão dos resultados obtidos da execução dos casos de teste, discussão sobre a usabilidade, otimização do carregamento e teste de carga e performance. A conclusão é dedicada a relembrar manutenção e melhoramentos para fases futuras do trabalho. É também divulgada uma ilustração do protótipo sobre a plataforma futura baseada neste trabalho.

ABSTRACT

The ubiquity and diversity of devices available in the market is changing the way we access and share information these days, being necessary to offer experiences of consistent use for a growing number of contexts.

This report describes the project "Pratos em Cartaz", developed under the course of the Master of Software Engineering. It is a platform to search for featured meals in restaurants near your location and to assist you to go there. A contextualization of the project is made and some data about the company involved in the project is presented. Also presented are some definitions and concepts, and an analysis of the state of the art on current key responsive frameworks is taken. The Project's problem analysis is described, requirements are presented, use case model and the domain model are also described. The development of the project is presented starting from the architecture of the solution, its interface and the technologies involved. Finally the report reviews and discusses the results obtained from the execution of validation test cases, usability discussion, optimization of loading and load performance testing. The conclusion is dedicated to recall maintenance and improvements for future phases of work. Also disclosed is an illustration of the prototype of the future platform based on this work.

March 2014

CONTEÚDO

1. Introdução.....	1
1.1 Contextualização	1
1.2 Apresentação da Empresa.....	1
1.3 Apresentação de objetivos do Projeto.....	4
1.4 Estrutura do documento	4
2. Definições e Conceitos.....	5
2.1 Introdução	5
2.2 Escolha de Desenvolvimento: Aplicação Nativa, Multiplataforma, ou Aplicação Web	8
2.3 Web Design Responsivo, Multidispositivo	9
2.4 Tecnologias e conceitos	11
2.5 Notas Finais	14
3. Análise do estado da arte	15
3.1 Introdução a Frameworks para aplicações responsivas	15
3.2 HTML5 Boilerplate.....	16
3.3 Zurb Foundation.....	18
3.4 Skeleton.....	21
3.5 HTML KickStart	22
3.6 YUI/Pure	23
3.7 Bootstrap.....	24
3.8 Comparação entre as frameworks estudadas	26
3.9 Notas Finais	27
4. Análise do Problema do Projeto	29
4.1 Introdução	29
4.2 Requisitos do projeto (com enunciado).....	29
Lista de Requisitos	30
4.3 Modelo de Casos de Uso	31
Diagrama de Casos de Uso:	31

Especificação dos Casos de Uso:	32
4.4 Matriz de Rastreabilidade Requisitos/Casos de Uso	35
4.5 Modelo de domínio	36
4.6 Notas Finais	37
5. Desenvolvimento do Projeto	39
5.1 Introdução	39
5.2 Arquitetura da solução	39
5.3 Base de Dados	40
5.4 Tecnologias envolvidas	41
5.5 Interface da solução	44
5.6 Notas Finais	49
6. Avaliação ou Discussão dos Resultados do Trabalho	51
6.1 Introdução	51
6.2 Testes de Validação	51
Caso de Teste 1	51
Caso de Teste 2	52
Caso de Teste 3	52
Caso de Teste 4	52
Caso de Teste 5	53
Caso de Teste 6	53
Caso de Teste 7	54
Caso de Teste 8	54
Caso de Teste 9	55
Caso de Teste 10	56
Caso de Teste 11	56
Caso de Teste 12	56
Caso de Teste 13	57
6.3 Testes de Usabilidade / Heurísticas de Nielsen	57
6.4 Otimização do carregamento da aplicação	59
6.5 Testes de Carga/Performance	60

6.6 Notas Finais	63
7. Conclusões e Trabalho Futuro	65
Referências.....	69

ÍNDICE DE FIGURAS

FIGURA 1: LOGOTIPO DA EMPRESA.....	2
FIGURA 2: HALL DE ENTRADA DAS INSTALAÇÕES DA EMPRESA	3
FIGURA 3: SALA UTILIZADA PARA AS REUNIÕES	3
FIGURA 4: VENDAS DE DISPOSITIVOS MÓVEIS JÁ SÃO SUPERIORES ÀS DE COMPUTADORES.....	5
FIGURA 5: ORIGEM DO TRÁFEGO NA INTERNET.	6
FIGURA 6: UTILIZAÇÃO MÓVEL DA WEB EM DISPOSITIVOS DE BOLSO - EXCLUINDO OS TABLETES.....	8
FIGURA 7: ILUSTRAÇÃO DE DESIGN WEB RESPONSIVO EM VÁRIOS ECRÃS.	10
FIGURA 8: VISUALIZAÇÃO EM RESOLUÇÃO DE PC DO <i>TEMPLATE</i> HTML5 BOILERPLATE.....	17
FIGURA 9: VISUALIZAÇÃO EM RESOLUÇÃO DE <i>SMARTPHONE</i> DO <i>TEMPLATE</i> HTML5BOILERPLATE.....	17
FIGURA 10: VISUALIZAÇÃO EM RESOLUÇÃO DE PC DO <i>TEMPLATE</i> ZURB FOUNDATION.....	18
FIGURA 11: EXEMPLO DE MENU DE NAVEGAÇÃO RECOLHIDO (À ESQUERDA), QUANDO VISUALIZADO EM RESOLUÇÃO DE <i>SMARTPHONE</i>	19
FIGURA 12: VISTA GERAL DOS COMPONENTES DE ZURB FOUNDATION.	20
FIGURA 13: CÓPIA DE ECRÃ COM VÁRIOS TEMPLATES PARA INICIAR UM PROTÓTIPO.	20
FIGURA 14: EXEMPLO DE PROTÓTIPO DE <i>TEMPLATE</i> PARA O ZURB FOUNDATION.	21
FIGURA 15: EXEMPLO DE ELEMENTOS DE UI.	22
FIGURA 16: EXEMPLOS DE ÍCONES DISPONÍVEIS.	23
FIGURA 17: SKIN BUILDER.	24
FIGURA 18: TEMA BOOTSTRAP PRE-COMPILADO DISPONÍVEL PARA COMEÇAR.....	25
FIGURA 19: DIAGRAMA DE CASOS DE USO.	31
FIGURA 20: MODELO DE DOMÍNIO ILUSTRANDO ENTIDADES E RELACIONAMENTOS.	36
FIGURA 21: MODELO DE DEPLOYMENT.	39
FIGURA 22: MODELO DE COMPONENTES.	40
FIGURA 23: ESQUEMA DA BASE DE DADOS.	41
FIGURA 24: ECRÃS DE ACOLHIMENTO NO ANDROID (À DIREITA COM O MENU ABERTO).....	45
FIGURA 25: INTERFACE DE VISUALIZAÇÃO DE SUGESTÃO DE PRATOS E SEU DETALHE.....	45
FIGURA 26: INTERFACE DE PESQUISA DE RESTAURANTES (À ESQUERDA) E DE PROXIMIDADE (À DIREITA).46	
FIGURA 27: INTERFACE DE VISUALIZAÇÃO DE UM RESTAURANTE.	47
FIGURA 28: INTERFACE DE VISUALIZAÇÃO DE UM RESTAURANTE	47
FIGURA 29: INTERFACE DO <i>BACK-OFFICE</i> DO RESTAURADOR PARA EDITAR LOCALIZAÇÃO E OS DADOS	48

FIGURA 30 INTERFACE DO <i>BACK-OFFICE</i> DO ADMINISTRADOR PARA INSERIR/APAGAR E ATIVAR/DESATIVAR RESTAURANTES	49
FIGURA 31: GOOGLE <i>DEVELOPERS PAGESPEED INSIGHTS</i> (ANÁLISE DE CARREGAMENTO DE PÁGINA WEB).	60
FIGURA 32: ANÁLISE DE TESTE DE STRESS DE UM CENÁRIO COM <i>LOADSTORM</i> PARA A ATIVAÇÃO DE DIÁRIA NO PRATOS EM CARTAZ.	61
FIGURA 33: ANÁLISE DE TESTE DE STRESS DE CENÁRIO COM <i>LOADSTORM</i> PARA A ATIVAÇÃO DE DIÁRIA NO PRATOS EM CARTAZ.....	62
FIGURA 34: TESTE DE CARGA COM O <i>PINGDOM TOOLS</i>	63
FIGURA 35 PROTÓTIPO FUTURO DE NOVA PLATAFORMA <i>WWW.EMREDOR.COM</i>	66

1. INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Este relatório foi elaborado no contexto de o meu projeto de Mestrado em Engenharia de Software, na Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Viana do Castelo. Resultado do estudo e trabalho efetuados por mim, e em resposta a um projeto proposto pela empresa **Mainstream, Centro de Inovação e Tecnologia Lda**, foi desenvolvida uma aplicação Web multi-dispositivo e responsiva para o setor da restauração.

Este documento relata a análise efetuada ao problema proposto, a análise do estado da arte no que respeita a *frameworks* para aplicações responsivas (*mobile-first*), descreve a arquitetura da solução, assim como as opções tomadas, e a aplicação desenvolvida com suas funcionalidades e interface, e no fim são avaliados os resultados obtidos através testes de validação (funcionais), de otimização de carga e performance, assim como é apresentada uma discussão sobre a usabilidade da aplicação.

1.2 APRESENTAÇÃO DA EMPRESA

A **Mainstream, Centro de Inovação e Tecnologia Lda** tem várias áreas de atuação, entre as quais a de desenvolver software para clientes e também a partir de ideias próprias com características inovadoras. As atividades da empresa são, entre outras, as seguintes:

- Desenvolvimento de Sistemas de Informação: Análise, arquitetura, implementação, segurança, gestão e manutenção;
- Formação Avançada: Informática, Gestão, Liderança, Formação qualificante, Formação específica à medida;
- Ninho de Empresas: Incubação, *Coworking* e Escritório Virtual, Acompanhamento, consultoria e serviços de apoio a *startups*, profissionais liberais e PME;
- Gestão de Sistemas de Informação: Especializada em serviços de auditoria e administração para empresas.

A empresa é dirigida pelo gerente Joaquim Campos Calhau, e tem por principal assistente a Engenheira Fabiana Regado. Estas duas pessoas são quem me têm recebido na empresa depois da rececionista à entrada. Foram-me apresentadas várias propostas de projeto, tendo escolhido, alguns dias depois da primeira entrevista, um dos projetos para desenvolver. O trabalho de mestrado tem sido conduzido sob orientação do

Professor Doutor Miguel Cruz, com o qual tenho reuniões semanalmente habitualmente às quartas-feiras.

Com a empresa têm sido agendadas reuniões, em geral semanalmente ou quinzenalmente, para eu poder apresentar o trabalho em curso, e assim podermos debater pormenores e melhorar os resultados. Estas reuniões servem sobretudo para eu obter *feedback* por parte da empresa e a aprovação do trabalho executado.

O projeto tem sido, portanto, desenvolvido segundo um modelo iterativo incremental – pequenos desenvolvimentos e obtenção de feedback – com ciclos de desenvolvimento extremamente curtos, normalmente quinzenal, como os métodos ágeis (ex.: SCRUM).

Os Contactos da empresa Mainstream, Centro de Inovação e Tecnologia Lda são:

Esposende, Rua Cidade São Domingos nº5.

Tel. 253 965 120 | Fax 253 966 047

Email: geral@mainstream.pt

Coordenadas GPS: 41.527248N, 8.777419W

Site Web: <http://www.mainstream.pt>

A Figura 1 apresenta o logotipo da empresa e as duas seguintes apresentam algumas fotos das instalações da empresa.



Figura 1: Logotipo da Empresa



Figura 2: Hall de entrada das instalações da empresa



Figura 3: Sala utilizada para as reuniões

1.3 APRESENTAÇÃO DE OBJETIVOS DO PROJETO

Os objetivos do projeto são o desenvolvimento de uma aplicação Web multi-dispositivo e responsiva para o setor da restauração, a qual deve funcionar em dispositivos móveis assim como em outros dispositivos, como o computador. Por fim deve ser elaborado o relatório de projeto de mestrado.

O tema do trabalho tem por ideia geral o seguinte enunciado:

«Implementar uma plataforma web onde todos os comerciantes de restauração possam aceder e diariamente carregar /modificar as suas ementas, de modo a que um cliente possa, no seu *smartphone*, pesquisar ofertas por restaurante ou mesmo procurar pela existência de determinado prato e apresentação das alternativas.»

1.4 ESTRUTURA DO DOCUMENTO

Este documento encontra-se dividido nos seguintes capítulos, para além do primeiro. O capítulo 2 apresenta Definições e Conceitos necessários para a leitura e compreensão do restante relatório. O capítulo 3 efetua uma Análise do Estado da Arte das *frameworks* para desenvolvimento de aplicações responsivas *mobile-first*. No 4º capítulo é efetuada a Análise do Problema e são apresentados artefactos de engenharia de software produzidos durante essa fase. O capítulo 5 relata o Desenvolvimento do Projeto e tece algumas notas sobre algumas opções tomadas. O capítulo 6 efetua a Avaliação e Discussão dos Resultados do Trabalho e, finalmente, o capítulo 7 apresenta Conclusões e Sugestões para Trabalho Futuro.

2. DEFINIÇÕES E CONCEITOS

2.1 INTRODUÇÃO

O ano de 2012 foi muito pouco comum no mercado do PC. Pela primeira vez desde 2001, as vendas de PCs foram mais baixas do que em relação ao ano anterior [1].

As vendas de *Tablets* ultrapassaram os 100 milhões de unidades no ano de 2012, e espera-se que tenham ultrapassado estes números no ano de 2013. A VisionMobile confirma estes dados muito recentemente no seu relatório de Fevereiro de 2014 [2]. Os *smartphones* são também um assunto em voga - de acordo com a Nielsen [1], a maioria dos assinantes de telefonia dos Estados Unidos agora possuem *smartphones*, e não possuem telemóveis tradicionais. Enquanto isso, a mudança para o móvel está a acontecer a uma velocidade extraordinária, tendo as vendas de *smartphones* ultrapassado os 600 milhões de unidades em 2012, em todo o mundo. A era *Post-PC* está a tomar forma e é real de facto desde de 2012 (ver Figura 4).

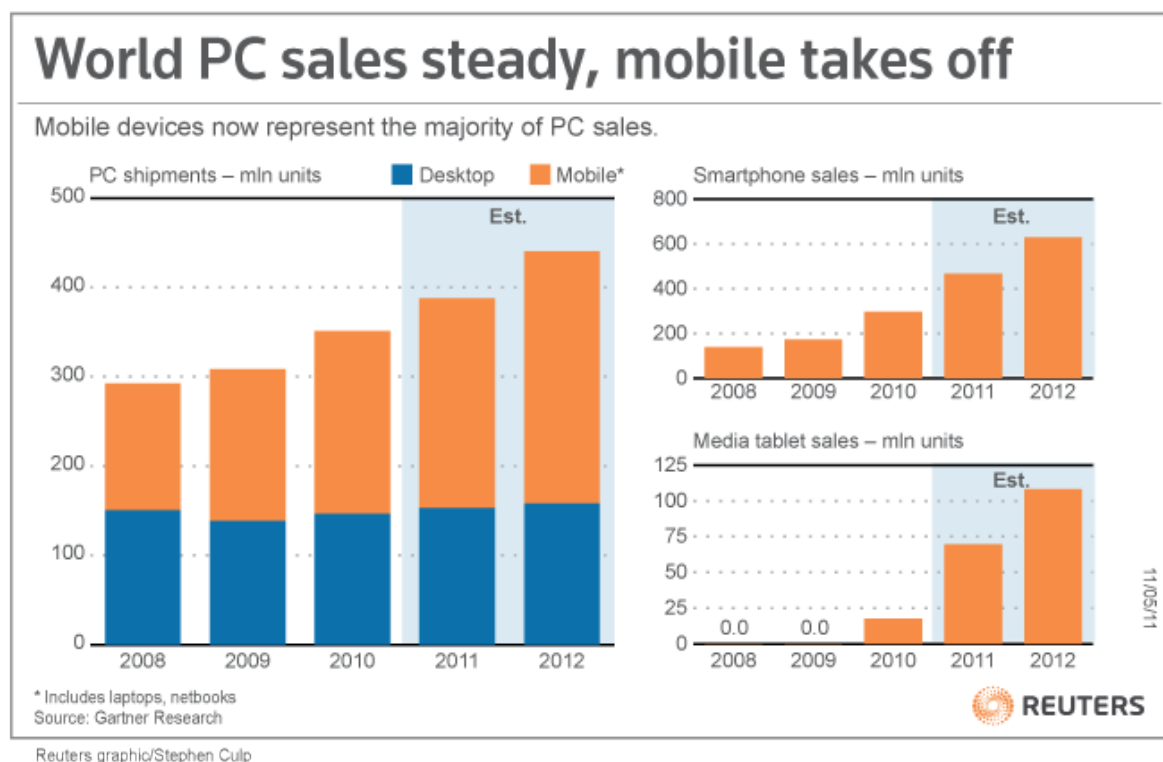


Figura 4: Vendas de dispositivos móveis já são superiores às de computadores

Fonte: <http://blog.thomsonreuters.com/index.php/tag/pc/>

Em termos de unidades, os *smartphones* são, entre os aparelhos com capacidade de ligação à Internet (aquilo a que a IDC¹ chama “equipamentos inteligentes conectados”), os que mais vendem: no mercado português, em 2013 foram postos à venda 4,12 milhões de aparelhos, dos quais mais 600 mil só no último trimestre. O ano passado foi aquele em que, tanto em Portugal como no resto do mundo, os *smartphones* ultrapassaram os telemóveis convencionais. Face a 2012, os *smartphones* cresceram 20,6% [3].

O tráfego na internet com dispositivos móveis também tem ganho importância. Por exemplo, as visitas com origem neste tipo de dispositivos a sites de atualidades como o *Mashable.com* vão de 30% a 50% até ao final do ano de 2013. Outra fonte, o *Hubspot.com* verificou 45% de visitas com origem em dispositivos móveis (incluindo *Tablets*) em Outubro-Novembro de 2013, contra 30% no ano anterior (ver Figura 5).

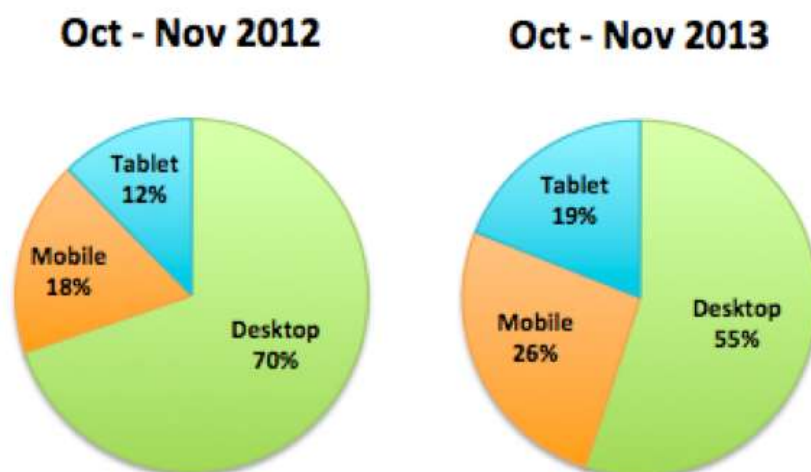


Figura 5: Origem do tráfego na Internet.

Fonte: <http://blog.hubspot.com/insiders/b2b-responsive-design>

Para aqueles de nós que criam *sites* e serviços, tudo isso leva a uma conclusão singular: milhões de ecrãs apareceram em 2013, e precisamos de construir aplicações para todos.

Desenvolver aplicações nativas para dispositivos móveis pode parecer a solução óbvia. Não há dúvida de que ter aplicações móveis para algumas das principais plataformas móveis é melhor do que não ter para nenhuma. E como desenvolvemos para cada *App Store* (*Apple, Google, Windows, Blackberry, Nokia, etc.*)? Certamente poderíamos criar aplicações nativas para um bom número destas plataformas, mas provavelmente não para todas. Existe também o exemplo das consolas de jogo com os mais variados modos de entrada e equipadas de browsers (*Nintendo, Play Station, etc.*).

¹ IDC é uma subsidiária da IDG - International Data Group, líder mundial na área dos media tecnológicos, estudos de mercado e de eventos.

Uma alternativa ao desenvolvimento de aplicações nativas para cada uma das plataformas é o desenvolvimento de aplicações “Web móvel”. Há dados que sugerem que a Web móvel é fundamental. De acordo com o *Pew Research Center* [1], 60% dos utilizadores de *Tablets* preferem a leitura de notícias na Web móvel do que através de uma aplicação. Enquanto quase metade dos adultos obtém notícias locais num dispositivo móvel, só 1 em 10 utiliza uma aplicação para este fim. Chamam-lhe o “*app gap*” (traduzo por: “fosso aplicacional”). Indica-se também na *Pew Research Internet Project* que 37% de proprietários de dispositivos móveis obtém dados sobre restaurantes e outros negócios locais no seu *tablet* ou *smartphone* [4].

Embora as empresas de *media* certamente devam oferecer aplicações nativas, parece claro que ter um excelente site móvel deve ser a prioridade para a generalidade das empresas. A solução óbvia é fazer com que um site funcione igualmente bem em todos os dispositivos. É neste contexto que entra a importância do **Web design responsivo** (RWD: *Responsive Web Design*) que ainda tem poucos anos de vida e que detalharemos mais à frente.

Luke Wroblewski [5] cunhou a frase “**Mobile-first**” (“Primeiro o Móvel”), em 2011. Na maioria das vezes, a experiência móvel para um site Web é desenhada e construída após uma versão completa para browser em PC. Em vez disso, Luke Wroblewski defende que há três razões para as aplicações web serem desenhadas primeiro para ambiente móvel:

- o móvel tem tido um forte aumento de uso;
- o móvel foca-se em concentrar conteúdo (reduzir àquilo que importa mais);
- o móvel estende as suas capacidades (geolocalização, movimento, etc.)

O espírito da época de desenvolvimento de software, pelo menos no que toca aos dispositivos móveis, de alguma forma tem vindo a desenvolver-se em direção a uma massa crítica de, não só “Primeiro o Móvel”, mas - como o CEO do Twitter afirmou recentemente: “All-in mobile” (“Tudo considerado em plataformas móveis”). Temos o exemplo da África saltando entre 1998 e 2005 para o *wireless* numa explosão de dispositivos móveis sem fios, sem passar pelas infraestruturas com fios a que nós europeus fomos habituados nas últimas décadas. O jornalista de renome Sul-Africano Toby Shapshak concluiu: “Simplificando, a África não é apenas um continente *Mobile-first*, é *Mobile-only*” (só móvel).

A Figura 6 apresenta um diagrama da utilização da Web móvel em dispositivos de bolso (excluindo os *tablets*) em Julho de 2012 e Julho de 2013. Reparámos que aproximadamente quase duplicou, sobretudo para algumas regiões como a África. Até na Europa passamos de 5,9% para 9,7%. É de esperar que a utilização da Web móvel terá duplicado ainda mais uma vez até os dias de hoje, desde de Julho de 2013 quando os dados foram recolhidos.

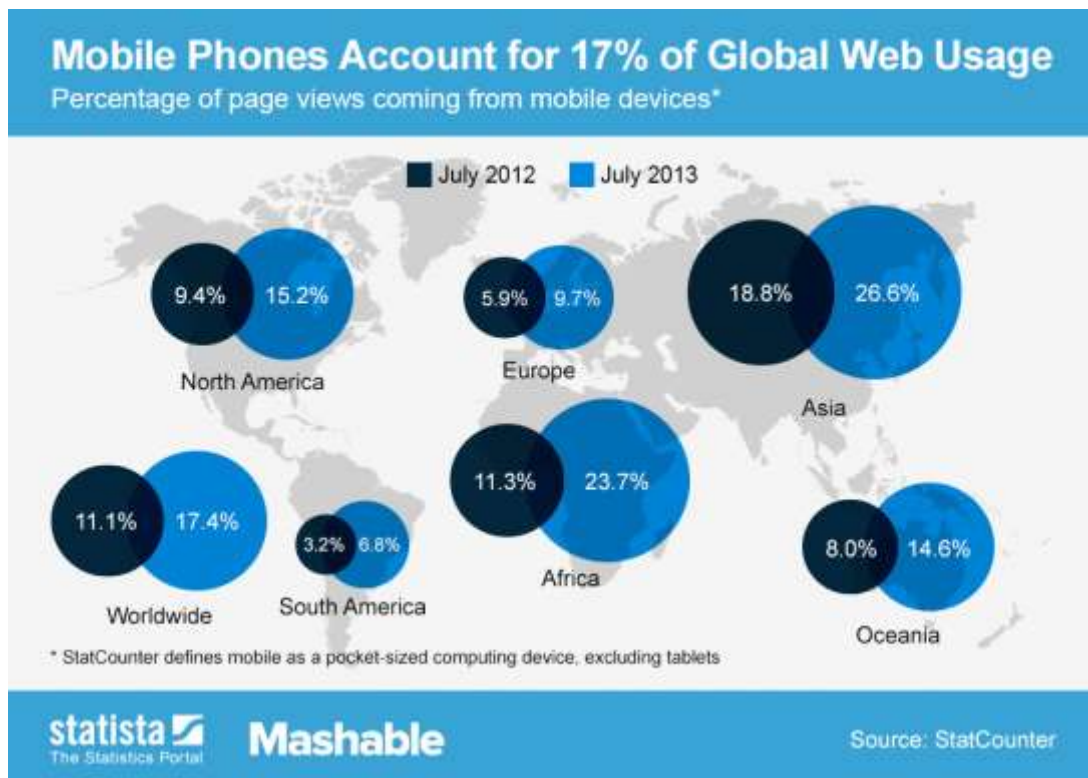


Figura 6: Utilização móvel da Web em dispositivos de bolso - excluindo os tablets.

Fonte: Statista.com e Mashable.com

É assim que depois de uma introdução com o destaque para a progressão súbita dos dispositivos móveis no mundo, prosseguimos com os conceitos para desenvolver software para estes dispositivos móveis.

2.2 ESCOLHA DE DESENVOLVIMENTO: APLICAÇÃO NATIVA, MULTIPLATAFORMA, OU APLICAÇÃO WEB

Como vimos o *tsunami* móvel já está em andamento. Como fazer? Quais são as opções se o nosso próximo projeto é uma aplicação móvel? O que isto significa para a nossa equipa? Quais as plataformas móveis que devemos concentrar-nos em apoiar? Para criar uma aplicação para um dispositivo móvel há geralmente três opções [6] [7]:

- Escrever a aplicação na plataforma/linguagem nativa do dispositivo.
 - Para *iOS*, isto significa escrever em *Objective C* usando um computador Mac, o *Xcode*, etc.
 - Para *Android*, isso significa escrevê-lo em *Java* usando *Eclipse* ou o seu *IDE* preferido, e o *Android SDK*, num computador Windows / Mac / Linux / etc.
 - Para o *Windows Phone / Windows 8*, isto significa escrevê-lo em *C# / XAML* em uma máquina *Windows 8 + Visual Studio 2012 / 2013*, etc.

- Usar uma "Ferramenta Multiplataforma ("Cross-Platform Tool - CPT").
Para citar VisionMobile[6], os CPTs podem ser usados para desenvolver aplicações nativas, híbridas e Web, e encontram-se em vários tipos de tecnologias: *Frameworks JavaScript*, *App factories*, *Web-to-native wrappers*, *Runtimes* e *Source code translators*. Uma das abordagens mais comuns é a aplicação móvel híbrida. Neste caso, a sua aplicação é, efetivamente, uma aplicação Web, mas hospedada dentro de um empacotador de aplicação nativa para o dispositivo. É instalada lançada e utilizada como qualquer aplicação nativa, e pode aceder *APIs* do dispositivo (com algumas exceções) mas é escrita em *HTML*, *JavaScript* e *CSS*. A abordagem mais popular é a de usar o *Apache Cordova* muitas vezes referida como *PhoneGap*, que é a redistribuição da marca *Adobe* de *Cordova*. *Telerik* e *Adobe*, têm ambas opções baseadas em *Cordova* para aplicações móveis híbridas. Existem outras abordagens CPT populares, incluindo *Titanium* (pertencendo à *Appcelerator*) e *Xamarin* (integrado por exemplo no *IDE Visual Studio*).
- Por último, mas não menos importante, temos a opção de **desenvolver um site Web móvel**. É uma aplicação Web, hospedada na Web e acedida pelo navegador do dispositivo móvel. Existe a possibilidade de criar uma versão separada e otimizada do site Web onde o utilizador é redirecionado quando está num *tablet* ou *smartphone*. E a outra possibilidade é de ser **Web responsiva**, que é um conceito ainda muito recente e inovador, razões pelas quais a preferimos além da satisfação pessoal e agilidade em usar certas tecnologias Web. Há outras vantagens e uma delas é que seja através de email, de redes sociais, de páginas Web, se temos informação útil online, as pessoas encontram-na e partilham-na com *links* [8].

Prosseguimos com o Web design responsivo que é a nossa escolha para o projeto proposto.

2.3 WEB DESIGN RESPONSIVO, MULTIDISPOSITIVO

O "Web design responsivo" (RWD, de *Responsive Web Design*) começou com um artigo de Marcotte em Maio de 2010, que cunhou o termo que refere-se a sites Web que mudam e adaptam a sua aparência para uma visualização otimizada em todos os tamanhos de ecrã, *tablets*, *smartphones*, *ipods*, *kindles* junto com ecrãs de computadores *desktop* e *laptop* e até televisores ou ecrãs de tamanho superiores. Um design serve para todos os ecrãs. Ocasionalmente, no setor de artes digitais, é também chamado de "*fluid design*" ou "*adaptive website design*"[9].

O RWD tenta adaptar o design ao tamanho do ecrã do dispositivo utilizado e requer a utilização de 3 noções fundamentais para a sua implementação[10]:

1. *Layouts* flexíveis, baseados em *grids* (grelhas), com dimensões relativas.

2. Conteúdos de texto escalável, imagem e vídeo flexíveis, através de redimensionamento dinâmico.
3. *Media queries* e *media query listeners*

Esta combinação de *grids* fluidas, multimédia escalável e *media queries* criam uma solução poderosa para construir aplicações fluidas e responsivas[11].

Media queries são expressões que verificam determinadas condições da página, aplicando diferentes regras de CSS para diferentes cenários[12] (ver exemplos em §2.4). São utilizadas para mudar o layout em diferentes aparelhos sem mudar o conteúdo[13]. *Media query listeners* foram criadas pelo W3C e providenciam uma API para interpretar e responder às regras definidas nas *media queries*.

Para os editores, o Web design responsivo oferece a maneira mais simples para atingir os leitores através de uma multiplicidade de dispositivos. Para os utilizadores, o RWD assegura uma grande experiência em todas as variedades (e orientações) de ecrãs.



Figura 7: Ilustração de Design Web Responsivo em vários ecrãs.

Exemplos de larguras de ecrã:

- 320 pixels - *Smartphones* no modo retrato.
- 480 pixels - *Smartphones* no modo paisagem.
- 600 pixels - *Tablets* pequenos. Ex: Amazon Kindle (600×800)
- 768 pixels - *Tablets* maiores em modo retrato. Ex: iPad (768×1024)
- 1024 pixels - *Tablets* maiores em modo paisagem, monitores antigos.
- 1200 pixels - Monitores *wide*.

2.4 TECNOLOGIAS E CONCEITOS

Nesta secção relembramos conceitos como *HTML*, *CSS*, *jQuery*, *DOM*, *ORM* entre outros que consideramos essenciais para compreender o restante documento.

O HTML (abreviatura para a expressão inglesa HyperText Markup Language) é uma linguagem utilizada para produzir páginas na Web. Os documentos HTML são interpretados por navegadores Web (*Web browsers*).

O CSS (Cascading Style Sheets) serve para apresentar e formatar o conteúdo HTML na página Web. É uma linguagem de estilo e permite a separação entre a forma e o conteúdo de um documento Web.

O Javascript é uma linguagem de script para poder, por exemplo, executar ações dinamicamente nas páginas do lado do navegador (cliente).

O jQuery é uma biblioteca JavaScript *cross-browser* desenvolvida para simplificar os scripts *client-side* que interagem com o HTML[14]. Foi lançada em janeiro de 2006 no BarCamp de Nova York por John Resig. Usada por cerca de 55% dos 10 mil sites mais visitados do mundo, jQuery é a mais popular das bibliotecas JavaScript. Permite ter uma camada virtual útil por cima da camada HTML/CSS/Javascript, e facilita a resolução da incompatibilidade entre os navegadores. Também permite a reutilização do código através de *plugins* criados por outros programadores. Fazendo uso de tais facilidades, os programadores podem criar camadas de abstração para interações de mais baixo nível, simplificando o desenvolvimento de aplicações Web dinâmicas. O jQuery funciona com AJAX e DOM e permite criar animações, manipular eventos e desenvolver aplicações AJAX.

O AJAX (Javascript Assíncrono e XML) é o uso metodológico de tecnologias como Javascript e XML para tornar as páginas Web mais interativas no *browser* para o utilizador, usando solicitações assíncronas de informações. Foi inicialmente desenvolvido por Jessé James Garret e mais tarde por diversas associações[15]. Apesar do nome, a utilização de XML não é obrigatória (JSON é frequentemente utilizado) e as solicitações também não necessitam de ser assíncronas. Isto para dizer que o AJAX é imprescindível e útil para carregar informação numa página Web sem solicitar o recarregamento desta no navegador e deixando o utilizador servir-se da página Web entretanto.

O DOM (*Document Object Model* - Modelo de Objetos do Documento) é uma especificação da W3C (World Wide Web Consortium: principal organização da World Wide Web para criar *standards*) para uma interface independente de plataforma e linguagem, onde pode-se dinamicamente alterar e editar a estrutura, conteúdo e estilo de um documento eletrónico, permitindo que o documento seja mais tarde processado e os resultados desse processamento, incorporados de volta no próprio documento[16]. A API DOM oferece uma forma padrão de aceder aos elementos de um documento.

Falamos também de PHP, cujo significado inicial era “Personal Home Page”, mas que a sua enorme evolução alterou para “PHP: Hypertext Preprocessor”, é uma linguagem de programação de ampla utilização, interpretada, que é especialmente interessante para desenvolvimento para a Web e pode ser mesclada dentro do código HTML. A sintaxe da linguagem lembra C, Java e Perl, e é fácil de aprender. O objetivo principal da linguagem é permitir a programadores escreverem páginas que serão geradas dinamicamente e rapidamente[17]. É uma linguagem de *scripting server-side Open Source* de uso geral, muito utilizada e especialmente formada para o desenvolvimento de aplicações Web incorporadas dentro do HTML.

As bases de dados são coleções organizadas de informações (dados) que se relacionam de forma a criar um sentido e dar mais eficiência durante uma pesquisa ou estudo[18]. O modelo de dados relacional baseia-se no princípio de que todos os dados estão guardados em tabelas (ou, matematicamente falando, relações). A linguagem padrão para as bases de dados relacionais é o SQL. Atualmente o SQL é adotado, apesar de suas restrições, porque é antigo, tendo ao longo do tempo demonstrado a sua utilidade e eficiência, e muito mais popular que qualquer outra linguagem de base de dados. São executadas transações numa base de dados. Uma transação é um conjunto de operações executadas numa base de dados, que deve ser visto, pelo sistema e pelo utilizador, como uma única ação. A integridade de uma transação depende de 4 propriedades, conhecidas como ACID (Atomicidade, Consistência, Isolamento, Durabilidade).

O Mapeamento objeto-relacional - ORM (Object-relational Mapping) - é outro conceito útil para o desenvolvimento. É uma técnica de desenvolvimento utilizada para reduzir o esforço da programação permitindo utilizar bases de dados relacionais de uma forma orientada aos objetos. As tabelas da base de dados são representadas através de classes e os registos de cada tabela são representados como instâncias das classes correspondentes. Com esta técnica, o programador não necessita de preocupar-se com os comandos em linguagem SQL; ele irá usar uma interface de programação simples que faz todo o trabalho de persistência de dados[19].

Voltamos às folhas de estilos CSS mencionando que o segredo do Web design responsivo está nas *media queries* CSS, fazendo variar o estilo consoante os ecrãs servidos. Usando *media queries*, as apresentações de páginas Web podem ser adaptadas para uma gama específica de dispositivos de saída sem alterar o conteúdo em si. Seria impraticável criar uma *media query* para cada resolução existente — seriam milhares! Devemos usar poucas *media queries* e fazer o *layout* fluir entre os *breakpoints*. O segredo está também em não usar pixels nas medidas, mas dar prioridade a um *layout* mais fluído e flexível baseado em unidades com percentagem “%” ou medidas tipográficas como o “EM” que são variáveis como a percentagem. Felizmente, o W3C criou os *media queries* como parte da especificação CSS3[12], melhorando com a promessa de novos tipos de media. Uma *media query* permite ter por alvo não apenas

certas classes de dispositivos, mas realmente inspecionar as características físicas do dispositivo que está a render o nosso trabalho [20].

Exemplo de código CSS com regra @media:

```
<style type="text/css">
@media screen and (max-device-width: 599px) {
    .column {
        float:none;
        width:100%; // toda a largura
    }
    Img, video { max-width: 100%; } //Não ultrapassa o container ...
}

@media screen and (min-device-width: 600px) and (max-device-width:
1023px) {
    .column {
        width: 50%; //metade da largura
        float: left;
    }
    Img, video { max-width: 100%; } //Não ultrapassa o container ...
}
</style>

<link rel="stylesheet" href="css/layout.css" media="all and (min-
width:1024px)">
```

Neste exemplo o código CSS assegura que as colunas estão apresentadas umas abaixo das outras (*float:none; width:100%*), isto para dispositivos pequenos, com 599 pixéis de largura máxima (é um *breakpoint*). Para dispositivos um pouco maior, largura de 600 ou 800 pixéis de um pequeno tablete por exemplo, as colunas serão apresentadas aos pares uma ao lado da outra (*width:50%; float:left;*). Para os ecrãs grandes acima de 1024 pixéis um ficheiro CSS exterior é fornecido neste exemplo. Considera-se geralmente mais *breakpoints* de resoluções como por exemplo as de 480 pixéis (dispositivo de bolso em modo *Portrait* e *Landscape*), 1200 pixéis (ecrã largo de PC) e 1920 pixéis (HDTV).

Os atributos *min-device-width* e *max-device-width* referem-se à resolução gráfica do dispositivo. Enquanto que podemos usar *min-width* e *max-width* que se referem à largura do *Browser*. Pode não ser o mesmo porque o *Browser* pode não estar maximizado no dispositivo.

2.5 NOTAS FINAIS

Depois de explicar alguns conceitos e definir algumas tecnologias inerentes ao projeto, a nossa escolha é construir algo que funciona em tantos dispositivos quanto possível. Esta opção irá fornecer uma organização melhor para concentrar recursos e orçamento sobre o conteúdo, em vez de manutenção. Se temos um site Web responsivo, quase não precisamos de mais uma aplicação móvel. E existe a vantagem da SEO (*Search Engine Optimization*) ser significativamente poderosa para divulgar o conteúdo de cada página Web nos motores de busca se for desejado. Poderá sempre ser criada uma aplicação nativa com um atalho para abrir o site Web móvel em plataformas específicas e divulgá-lo normalmente em cada App Store visada. Não é necessário uma equipa de programadores especializados em várias plataformas. Pondera-se assim poupar em recursos humanos sobretudo se a equipa é mínima. E a manutenção é concentrada numa plataforma única Web. Assim o RWD pareceu-me a melhor escolha neste projeto onde seria *a priori* o único programador, teria que desenvolver uma aplicação para diferentes plataforma móveis e não móveis, e onde teria livre arbítrio de seleção das tecnologias para iniciar e levar a bom termo o projeto.

3. ANÁLISE DO ESTADO DA ARTE

3.1 INTRODUÇÃO A FRAMEWORKS PARA APLICAÇÕES RESPONSIVAS

O conceito de design responsivo não é uma grande novidade, mas tem chamado bastante atenção nos últimos tempos com o crescimento do mercado de dispositivos móveis, principalmente por conta dos *tablets* e *smartphones* de tamanhos e resoluções cada vez mais variados. Neste novo contexto dos navegadores e de várias resoluções, o design responsivo surge como uma evolução lógica do design de sites, também conhecido como Web design [21]. Anteriormente, grande parte da internet era acedida por resoluções e navegadores muito semelhantes. Até há pouco tempo atrás bastava fazer um site que funcionava em Internet Explorer com resolução máxima de 1024×768 pixels e tudo estava resolvido. Claro que existiam outras características, mas a grande maioria estava nesse mesmo grupo. Hoje tudo mudou, temos TVs de 50" polegadas a aceder à internet, temos telemóveis com ecrãs de 2" até 5", *tablets* de 6" até 11" polegadas e sem contar os próprios computadores, com ecrãs de *netbook* reduzidos até aos mais novos iMacs da Apple, colocando uma margem de 11" até 27" polegadas. Esta característica demonstra que um site pode ser visto de diversas formas e em diversos contextos, e é para isto que os sites devem estar preparados. Teríamos provavelmente que fazer inúmeros sites Web diferentes para satisfazer as diferentes combinações de tamanhos e resoluções de ecrãs. A evolução do design responsivo mudou bastante tudo isto, conseguindo trazer a vantagem de ter diversos sites Web num só lugar. O design responsivo, como o próprio nome já indica, consegue responder ao tamanho e resolução do ecrã para se adaptar da melhor forma. Em vez de criar dois sites separados, um para mobile e um para desktops, como é ainda muito comum, usando RWD pode-se criar apenas um site que vai se adaptar muito bem a qualquer ecrã onde for carregado o site Web.

Como os navegadores interpretam os sites?

Em primeiro lugar há o *user agent*, utilizado para diferenciar o utilizador quando está a aceder através de um *smartphone* ou do navegador de *desktop*. É a função dentro do site que identifica as questões técnicas envolvidas no utilizador que está visitando o site.

Exemplo de *User Agent* no iPad com Firefox:

```
Mozilla/5.0 (iPad; U; CPU OS 3_2_1 like Mac OS X; en-us)
AppleWebKit/531.21.10 (KHTML, like Gecko) Mobile/7B405
```

Este exemplo retorna o navegador, sistema, língua, etc. utilizados pelo utilizador. Há aqui um problema, é que muitos dispositivos e navegadores podem não estar incluídos na lista extensa de navegador/sistema (i.e. um caso excecional) do script *Javascript* que deteta estas características; portanto as páginas Web não seriam redirecionadas para o URL adequado do site onde seria apresentado um conteúdo personalizado para este

tipo de configuração de dispositivo com um dado sistema e um dado navegador esperados ou não.

Em segundo lugar, em paralelo, os “*media queries* e seus *listeners*” são uma propriedade do CSS que identifica qual a resolução de ecrã onde o site está a ser carregado e serve determinado estilo. O design responsivo atual funciona basicamente com esta função, modificando tamanhos, dimensões e formatos conforme o tamanho do ecrã do utilizador. Por exemplo se for redimensionada a janela do browser, quando esta função do CSS identificar uma largura de janela diferente, isto poderá mudar o visual do site, sem ter nenhuma alteração no conteúdo carregado, apenas na forma como está apresentado o site responsivo.

Esta técnica de design já é defendida por muitos investigadores há algum tempo, mas só recentemente começou a ganhar destaque depois de a empresa Google Inc. confirmar indiretamente que para a área de *Search Engine Optimization* (SEO) o design responsivo é o mais indicado entre as soluções de desenvolvimento para o Web móvel[22]:

«Google recommends webmasters follow the industry best practice of using responsive web design, namely serving the same HTML for all devices and using only CSS media queries to decide the rendering on each device... We recommend using responsive web design because it has many good aspects...»

Portanto o Google recomenda o uso de design Web responsivo, e já está a desenvolver os seus algoritmos nesta direção para a interpretação do conteúdo das páginas Web. Parece que é uma vantagem escolher este tipo de desenvolvimento RWD naquilo que toca à SEO.

Vamos ver algumas *frameworks* que funcionem com design responsivo. As *frameworks* RWD são estruturas de código já prontas que trazem com elas diversas linhas em CSS, *Javascript* e *HTML* já escritas, facilitando bastante o trabalho do programador do site.

3.2 HTML5 BOILERPLATE

Segundo o respetivo site Web, HTML5 Boilerplate é um *template front-end* profissional para desenvolver rapidamente aplicações ou sites Web robustos e adaptáveis. O seu uso é muito popular (ver *Github stars*), certamente porque está a ser desenvolvido por centenas de programadores há vários anos (desde 2010). Ele não impõe uma filosofia ou *framework* de desenvolvimento específico, pelo que somos livres de escolher a arquitetura do nosso código da forma que queremos.

Dados descritivos:

Homepage: <http://html5boilerplate.com>

Código fonte: <https://github.com/h5bp/html5-boilerplate>

Github Stars: 23 713

Github Forks: 5 991



Figura 8: Visualização em resolução de PC do *Template* Html5 Boilerplate.



Figura 9: Visualização em resolução de *smartphone* do *Template* Html5Boilerplate.

Foi feito pensando em ícones, em *Google Analytics* e mais, como por exemplo:

- Um *Template* HTML leve e *Mobile-friendly*;
- Inserção de código otimizado para o *Google Analytics*;
- Implementação para ícones *Touch-device*;
- Documentação extensa cobrindo dúzias de itens extra para desenvolver com esta *framework*, como por exemplo as configurações do servidor Apache para ajudá-lo a proporcionar uma excelente performance do site (são fornecidos dezenas de ficheiros como, por exemplo, um ficheiro de configuração *.htaccess* para o servidor).

Pareceu-me uma solução robusta, popular e certamente a mais antiga, mas pareceu-me também bastante complexa para começar a usar.

3.3 ZURB FOUNDATION

Zurb é uma empresa de design de interação e de estratégia de design, com sede em Campbell, na Califórnia. Segundo o site Web <http://foundation.zurb.com>, Foundation seria o *framework front-end responsive* mais avançado do mundo inteiro. Com Foundation pode-se desenvolver protótipos rapidamente e criar sites ou aplicações funcionando em qualquer dispositivo, com toneladas de *layouts* construídos (i.e. uma grelha responsiva), elementos como o *orbit* que é um *slide* de imagens, etc. (ver Figura 10).

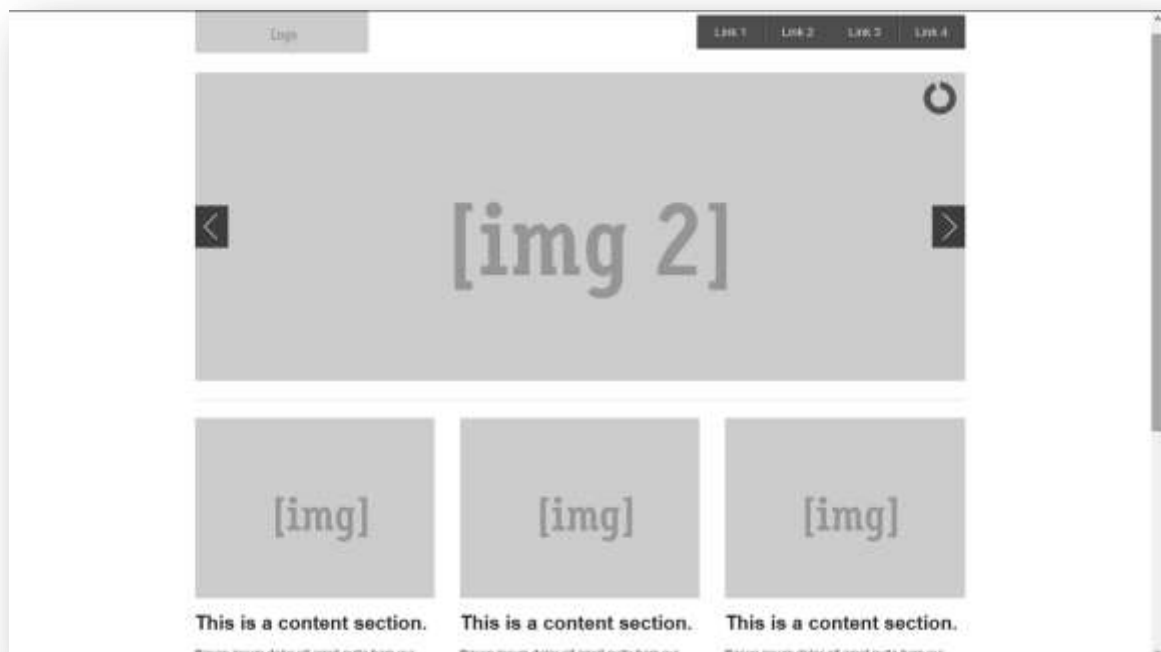


Figura 10: Visualização em resolução de PC do Template Zurb Foundation.

URL da Fonte: <http://foundation.zurb.com/templates/banded.html>

Verifiquei também que estavam disponíveis Templates com elementos interessantes como por exemplo um menu retraído em janelas de pequena dimensão:

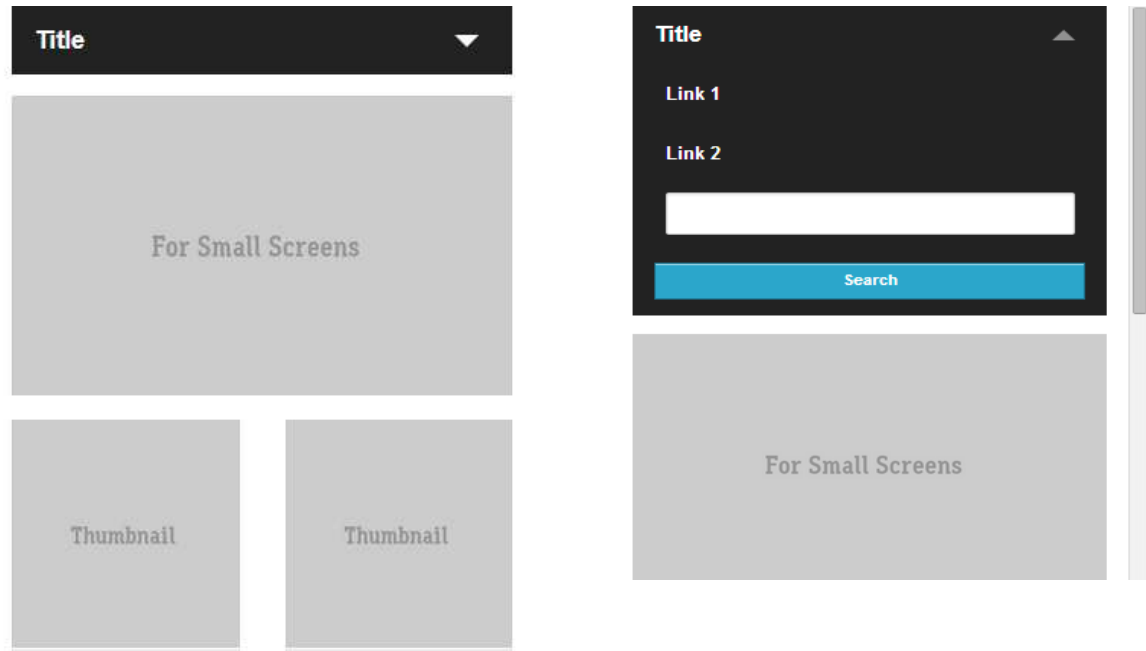


Figura 11: Exemplo de menu de navegação recolhido (à esquerda), quando visualizado em resolução de *smartphone*.

Fonte: Download dia 03/02/2014 do tema no URL: <http://www.themplio.com/themes/free-foundation-templates>

Este tipo de *layout* aparece-nos com um design simples e minimalista, ótimo para poder começar a desenvolver se acharmos este tipo de *Template* adaptado àquilo que se quer para a nossa aplicação Web. É um bom ponto de partida para arrancar rapidamente com um projeto Web de design responsivo.

Na figura seguinte (Figura 12) podemos ficar com uma ideia geral de alguns elementos da interface Zurb Foundation.

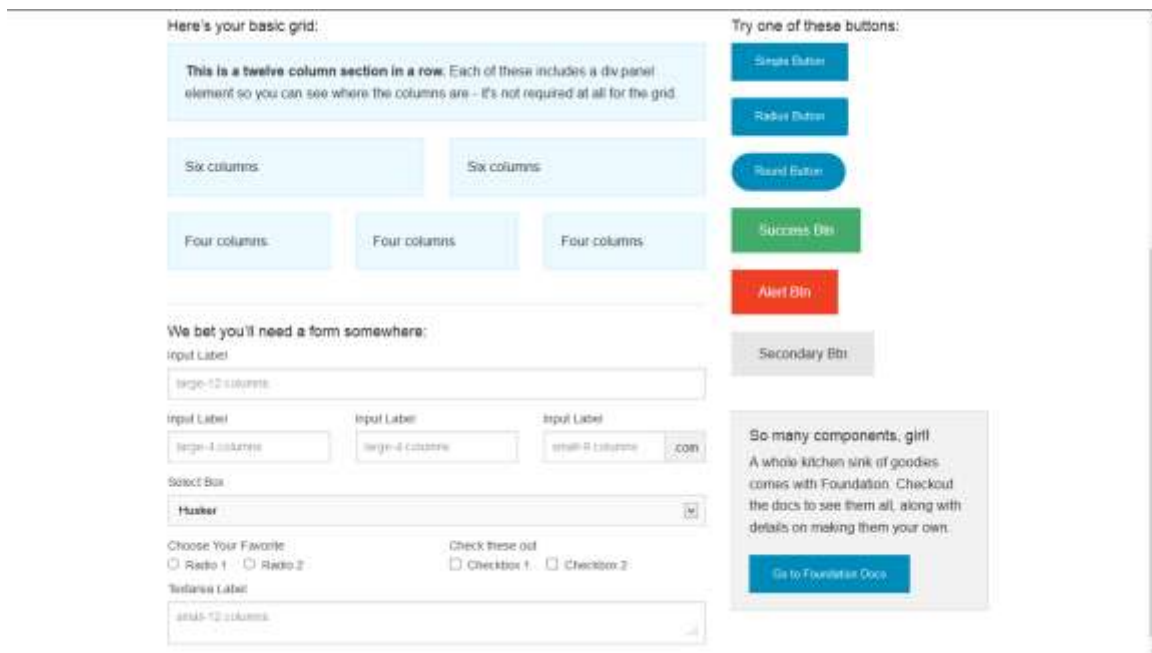


Figura 12: Vista geral dos componentes de Zurb Foundation.

Podemos consultar o link seguinte: <http://foundation.zurb.com/templates.html> para ter uma ideia de como desenvolver um protótipo de interface seguindo um *Template* de base (ver as 2 imagens abaixo Figura 13 e Figura 14).

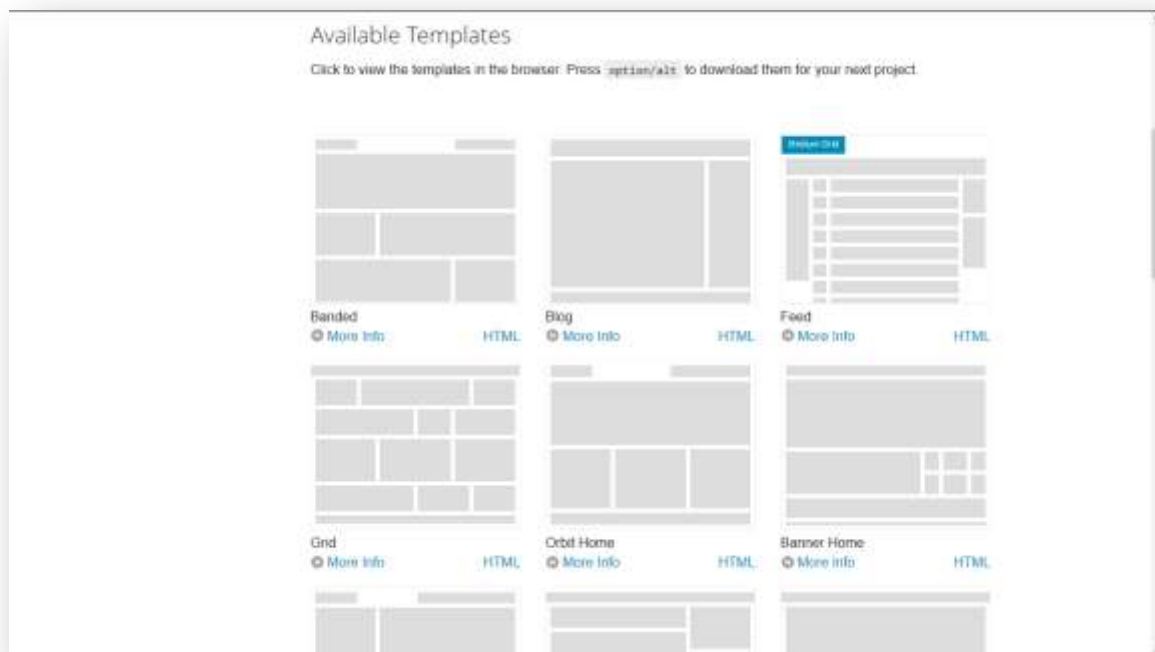


Figura 13: Cópia de ecrã com vários Templates para iniciar um protótipo.

Fonte: <http://foundation.zurb.com/templates.html>



Figura 14: Exemplo de protótipo de *Template* para o Zurb Foundation.

URL de cópia de ecrã: <https://www.notableapp.com/posts/f5c3a40dc27421a7ae447873c1bc4ca129aff9a1#annotate>

Dados descritivos:

<https://github.com/zurb/foundation>

Github Stars: 15 627

Github Forks: 3 319

Pareceu-me um projeto com muita ambição, mas com muita consistência. Várias fases avançadas de desenvolvimento de um protótipo gráfico podem ser consideradas de um ponto de vista muito profissional a partir de numerosos *Templates* de base. Trata-se de uma solução muito avançada, se for desejada, e adequada preferivelmente para projetos de grandes dimensões, na minha opinião.

3.4 SKELETON

Segundo o próprio site Web (www.getskeleton.com) Skeleton (versão 1.2) é um conjunto bem organizado de ficheiros CSS que ajudam a desenvolver rapidamente sites Web com um visual ótimo em qualquer tamanho de ecrã, seja num 17" polegadas de PC ou num iPhone. É baseado numa grelha responsiva, mas fornece também CSS básico para tipografia, botões, formulários e *media queries*. Dá o ênfase que é muito leve com muito poucos KiloBytes para a dimensão dos ficheiros necessários.

É *Open Source* e pode-se encontrar o *download* em: <https://github.com/dhg/Skeleton>.

Está presente no arquivo *downloaded*:

- um ficheiro *index.html*
- 3 ficheiros CSS (*base.css*, *skeleton.css*, *layout.css*)
- uma pasta *images* com um ficheiro *favicon.ico* 16x16 e outros 3 ficheiros *apple-touch-icon* para as 3 dimensões de ícones Apple (iPhone 3, iPad, iPhone 4 retina).
- Está também disponível para *download* separado um ficheiro *Skeleton-grid.psd* (com o desenho de colunas numa grelha vazia de conteúdo no caso de um designer gráfico querer explorar).

Tem 4175 Stars no GitHub e 789 Fork.

Achei que havia ainda pouca documentação para arrancar rapidamente com o desenvolvimento de um site Web responsivo.

3.5 HTML KICKSTART

Segundo o site Web (<http://www.99lime.com/elements/>) o HTML KickStart é um conjunto de ficheiros *HTML5*, *CSS*, *JS* ultraleve, *layouts*, e elementos desenhados para colocar rapidamente em produção um site Web e poupar dezenas de horas de trabalho num próximo projeto Web. Vai na versão 0.94.

A licença é MIT Open Source.

É *designer friendly* com 249 ícones, e toda a UI (*User Interface*) é vasta e otimizada para um design responsivo (exemplo: *Table sortable*, *Tooltips*, *Tabs*, *Menus* responsivos, etc.).

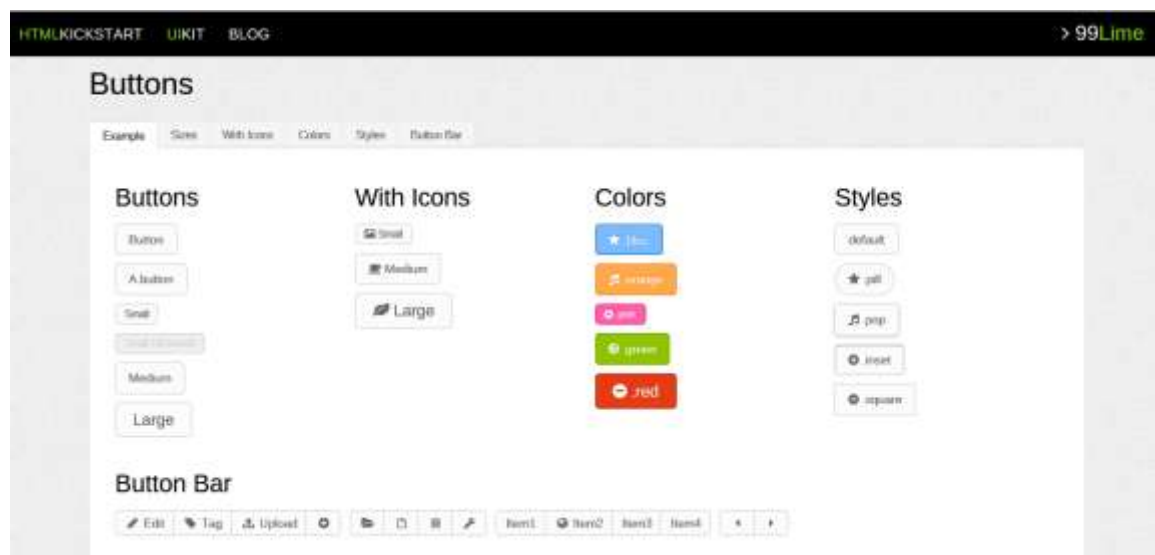


Figura 15: Exemplo de elementos de UI.

Fonte: <http://www.99lime.com/elements/#btn-example>

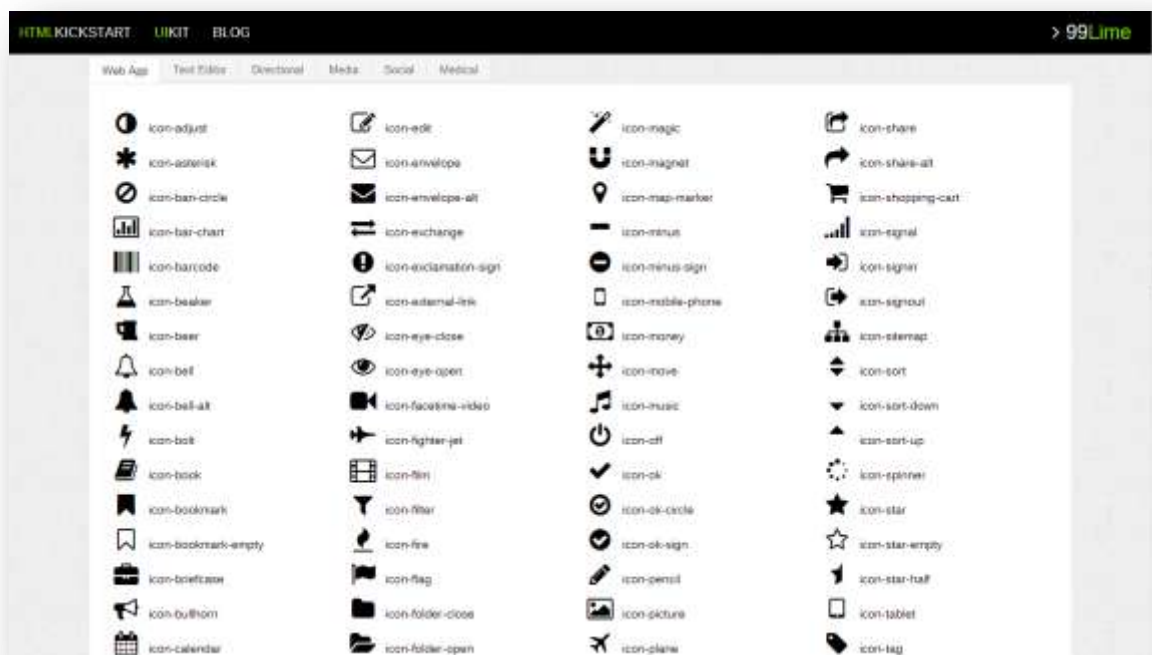


Figura 16: Exemplos de ícones disponíveis.

Fonte: <http://www.99lime.com/elements/#icons-web>

Site no *Github*: <https://github.com/joshuagatcke/HTML-KickStart>

Só tem *1.032 Stars no Github*, mesmo se o site www.99lime.com indica “*Downloaded over 166190 Times*”. Pareceu-me muito bom pela UI e as funcionalidades fornecidas (por ex.: *Table sortable*: para dispor os elementos de uma tabela por ordem alfabética), mas é pena ter pouca popularidade de facto.

3.6 YUI/PURE

Pure é um conjunto de pequenos módulos CSS responsivos que podem ser usados em todos os projetos Web (<http://purecss.io>). Ainda é muito recente e pertence ao grupo Yahoo YUI que é *um framework open source JavaScript e CSS* para criar aplicações Web interativas ricas.

Na Figura 17 podemos visualizar uma ferramenta (*Skin Builder*) disponível *online* para criar a folha de estilo CSS de uma aplicação Web.

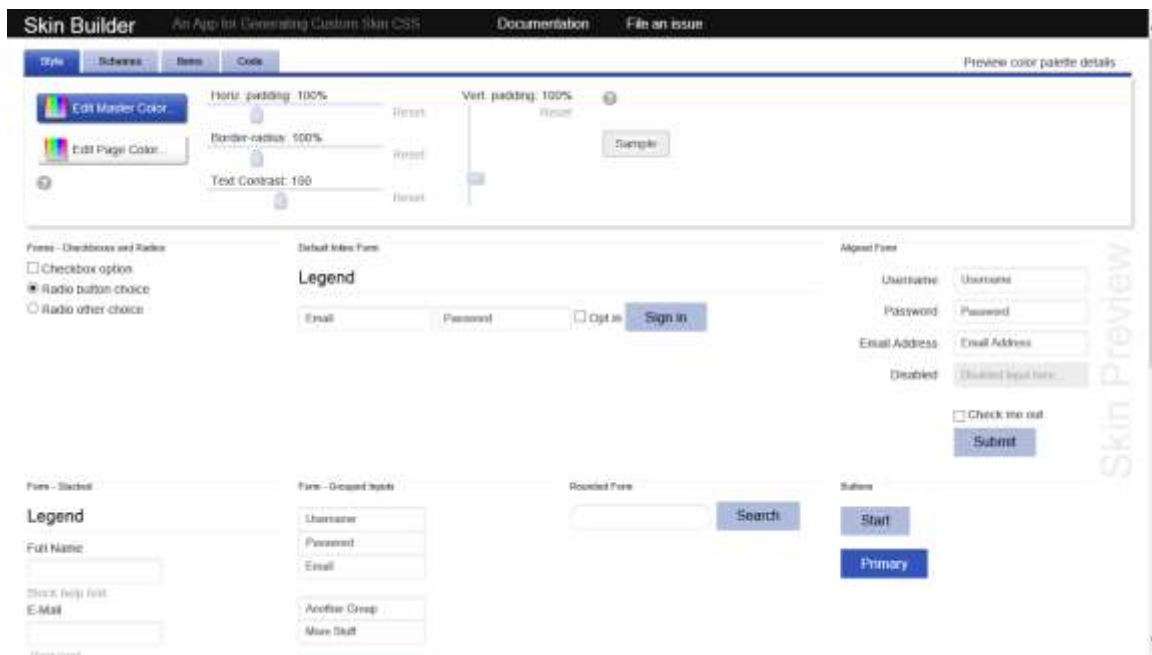


Figura 17: Skin builder.

Fonte: <http://yui.github.io/skinbuilder/>

Github: <https://github.com/yui/pure>

7234 Stars Github, 908 Fork

3.7 BOOTSTRAP

Segundo o próprio site Web (<http://getbootstrap.com> visitado a 11/03/2014) Bootstrap é o mais popular dos *framework front-end* para desenvolver projetos responsivos na Web, e *mobile-first desde a Versão 3*. “Desenhado para todos, e em todos os lugares”, Bootstrap torna o desenvolvimento Web *front-end* mais rápido e mais fácil. É feito para pessoas de todos os níveis, dispositivos de todas as formas e para os projetos de todos os tamanhos.

One framework, every device.

Bootstrap easily and efficiently scales your project with one code base, from phones to tablets to desktops.

Bootstrap foi desenvolvido por Mark Otto e Jacob Thornton no Twitter como uma *framework* para encorajar a consistência entre as ferramentas internas. Antes de Bootstrap, várias bibliotecas foram utilizadas para o desenvolvimento de interfaces, o que levou a inconsistências e um volume de manutenção elevado. Em agosto de 2011, o Twitter lançou o Bootstrap em *Open Source*. Em fevereiro de 2012, foi o projeto de

desenvolvimento GitHub mais popular. Já foi utilizado pela NASA (<http://spotthestation.nasa.gov>), MSNBC (<http://www.breakingnews.com>) entre outros.

Twitter Bootstrap é uma coleção gratuita de ferramentas para a criação de sites e aplicações Web. Contém HTML e modelos de design baseados em CSS para tipografia, formulários, botões, gráficos, navegação e outros componentes de interface (exemplo: ícones fornecidos gratuitamente pela www.glyphicon.com), bem como extensões JavaScript opcionais. Tem um sistema de grelha de 12 colunas e *plugins Javascript* notáveis, tais como janelas modais, *tooltips*, carrossel etc.

Bootstrap está disponível para *download* em duas formas, proporcionando ambas as variações ficheiros compilados e *minified*. A primeira é o *Precompiled Bootstrap*, pronto para começar. A outra é o *Bootstrap source code* cujo download inclui o *precompiled CSS*, *JavaScript*, e *font assets*, com o código fonte LESS, *JavaScript*, e a documentação.

Bootstrap é feito com LESS no seu núcleo. Sass e LESS são pre-processadores que permitem escrever CSS limpo numa programação construída em vez de regras estáticas. Stylus é outra linguagem de *stylesheet dinâmica* influenciada por Sass e LESS. É visto como a terceira sintaxe mais usada para pre-processadores CSS. A folha de estilo chamada *bootstrap.less* inclui os componentes de folhas de estilo. Os programadores podem adaptar os ficheiros do Bootstrap, e seleccionar os componentes que desejam usar no seu projeto. Pode-se também usar uma configuração online no site do Bootstrap nomeado de “Customize” (<http://getbootstrap.com/customize/>).

A Figura 18 apresenta um *Template* pronto para começar a desenvolver um site Web responsivo.

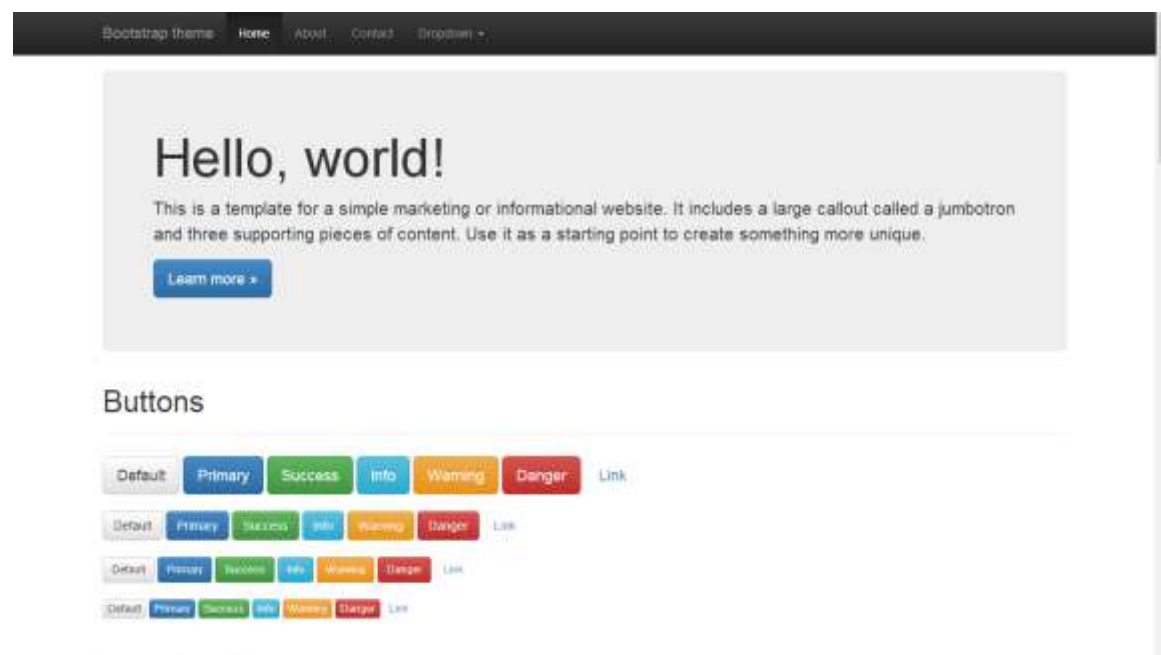


Figura 18: Tema Bootstrap pre-compilado disponível para começar.

Fonte: <http://getbootstrap.com/examples/theme/>

Github: <https://github.com/twbs/bootstrap>

Github Stars: 64 449 (no mês passado eram 53 668, portanto está a crescer muito)

Fork: 23147

Licenças:

- Código: Apache License v2.0
- Documentação: Apache License v2.0
- Ícones: CC BY 3.0 from Glyphicons Free

3.8 COMPARAÇÃO ENTRE AS FRAMEWORKS ESTUDADAS

Nesta secção é feita uma análise comparativa entre as várias frameworks de RWD apresentadas na secção anterior.

O resultado da análise comparativa é apresentado na grelha de comparação abaixo, tendo sido usados critérios como a popularidade da plataforma, medida através do Github Stars (número de interessados) e Github Fork (número de developers), facilidade de uso para o programador e *User Experience* da UI final.

A escala utilizada para classificação dos vários critérios de comparação é a seguinte:

Muito mau	Mau	Razoável	Bom	Muito bom
--	-	+	++	+++

Grelha de comparação:

Frameworks:	HTML5 Boilerplate	Zurb Foundation	Skeleton	HTML KickStart	YUI Pure	Twitter Bootstrap
Critérios:						
Github Stars (interessados)	+++ 24.052	++ 15.736	++ 4.175	+ 1.036	++ 7.238	+++ 64.457
Github Fork (developers)	+++ 6.125	+++ 3.361	+ 789	+ 226	+++ 908	+++ 23.152
Facilidade para Programação	- (ficheiros)	++	-- (doc)	++	-	+++
User experience da UI final	+	+++	-	+++	+	+++
Início do projeto	Janeiro 2010	Outubro 2011	Maio 2011	Abril 2013	Maio 2013	Agosto 2011
Versão atual	V 4.3.0	V 5.2.1	V 1.2	V 0.94	V0.4.2	V3.1.1
Versão lançada (visto 15/03/14)	Há 6 meses	Há 2 dias	Há 2 anos	Há 2 anos	Há 1 mês	Há 1 mês

É ainda apresentada na grelha acima, para cada Framework, a data de início do respetivo projeto de desenvolvimento e a versão atual com o seu último lançamento.

Justificamos as classificações atribuídas na grelha, da forma seguinte:

O HTML5 Boilerplate tem boas pontuações em termos de popularidade e longevidade de existência, mas peca pelo aspeto complexo para programar visto que utiliza ficheiros diversos para otimizar a performance como por exemplo um *“.htaccess”* e outros, e a UI não me pareceu extraordinária.

Quanto ao Zurb Foundation está bastante popular, com uma boa UI, mas é sobretudo dedicado a soluções de grande dimensão com extensas funcionalidades profissionais, é o que retira um pouco o lado simples de programação.

O *framework* Skeleton tem disponíveis muito poucos exemplos e pouca documentação, sem grandes destaques para a UI, tendo motivado a atribuir-lhe uma pontuação muito fraca em termos de facilidade de programação.

Considerando o HTML KickStart, ele tem uma UI muito boa, apresentando funcionalidades excecionais como por exemplo a possibilidade de ter tabelas que ordenem-se automaticamente. O ponto fraco seria a popularidade no Github e o tempo de vida sendo ainda um dos *frameworks* considerados dos mais recentes. Quanto ao YUI/Pure tem vindo a crescer em termos de popularidade mas é ainda muito recente também e ainda não achei muita facilidade em programar com esta *framework* em geral, mesmo se já existem online ferramentas para criar uma boa UI globalmente.

Observamos que o Twitter Bootstrap tem a pontuação máxima em todos os aspetos considerados. É de longe a mais popular, e é mais popular que todas as outras *frameworks* juntas. Estão disponíveis muitos *Templates CSS online* grátis para servir ao bom arranque de um projeto e o Bootstrap é considerado escalável, simples para projetos sejam eles grandes ou pequenos. A UI tem as suas qualidades e a programação desta é facilitada pelos numerosos exemplos na documentação *online*.

Portanto Bootstrap é o vencedor desta comparação.

3.9 NOTAS FINAIS

A plataforma escolhida para o desenvolvimento do projeto foi a que tem mais pontuação na grelha acima de comparação das várias *frameworks*. O Bootstrap da Twitter pareceu-me o valor mais seguro para investir tempo e conhecimento e levar a bem o projeto até ao fim. Tem muitas funcionalidades, documentação extensa *online* para facilitar o desenvolvimento, etc. Há também *Templates* grátis disponíveis em vários sites Web consideráveis para a base do design da nossa aplicação Web. O Bootstrap já tem uma certa maturidade porque é o segundo *framework* considerado mais antigo, existe desde de Agosto de 2011 e vai na sua versão 3.1.1, continuando ativo o seu projeto de desenvolvimento.

4. ANÁLISE DO PROBLEMA DO PROJETO

4.1 INTRODUÇÃO

Como mencionado anteriormente, 37% de proprietários de dispositivos móveis obtém dados sobre restaurantes e outros negócios locais no seu tablete ou *smartphone* [4]. Estes dados realçam a necessidade de aplicações móveis à volta da restauração e outros tipos de negócios.

O problema do trabalho “Pratos em Cartaz” foi analisado entre várias reuniões com a empresa Mainstream, em Esposende, para por exemplo levantar e discutir requisitos. O professor Doutor Miguel Cruz também deu orientação para analisar o problema entre várias reuniões comigo próprio. A ideia era chegar a uma aplicação funcional para tudo correr bem, e com a possibilidade posterior de poder acrescentar mais funcionalidades.

4.2 REQUISITOS DO PROJETO (COM ENUNCIADO)

Podemos ver aqui o enunciado completo para o projeto proposto «Pratos em Cartaz» pela empresa Mainstream:

A ideia geral:

- Implementar uma plataforma web onde todos os comerciantes de restauração pudessem aceder e diariamente carregar /modificar as suas ementas, de modo a que um cliente possa, no seu *smartphone*, pesquisar ofertas por restaurante ou mesmo procurar pela existência de determinado prato e apresentação das alternativas.

Destinatários:

- Comerciantes de restauração de Esposende, para começar, como produtores de informação;
- Público em geral, como consumidor da informação.

Tecnologias envolvidas:

- Servidor web;
- web mobile ou mobile *app*.

Benefícios:

- Divulgação das ementas e possível atração de mais clientes;
- Possível multilíngue servindo também para turistas de nacionalidade estrangeira.

Operacionalização:

- por protocolo com a ACICE. Custos de desenvolvimento/implementação a definir forma de os cobrir.

Outras notas:

- Possível contemplar a exibição de prémios/menções de determinado prato obtidos em concursos gastronómicos;
- ligação a googlemaps ou mesmo a navegação para mostrar direções;
- Cada restaurante poderia incluir para a sua página elementos como foto, meios de pagamento, contatos, etc..

Enquadramento:

- desenvolvimento do concelho;
- turismo;
- modernização do comércio.

Após discussão e, como combinado com a empresa, relegando o problema do multilíngue e os prémios gastronómicos para uma versão futura, elaborámos os requisitos seguintes:

LISTA DE REQUISITOS (ENUMERAMOS AS AÇÕES POSSÍVEIS PELOS 4 ATORES CONSIDERADOS):

Id	Requisito
R1	O Restaurador Proponente regista-se no site Web.
R2	O Administrador valida ou rejeita a inscrição do restaurador proponente.
R3	Quando o Administrador valida ou rejeita a inscrição, o sistema envia email com Login e Password para o restaurador proponente e insere na BD os dados para o novo restaurante.
R4	O Administrador pode proceder à ativação/desativação da conta do restaurador e restaurante.
	O Restaurador pode no <i>back-office</i> :
R5	<ul style="list-style-type: none"> • fazer <i>login</i>
R6	<ul style="list-style-type: none"> • fazer <i>logout</i>
R7	<ul style="list-style-type: none"> • inserir prato
R8	<ul style="list-style-type: none"> • editar prato (Disponibilidade, Descrição, Preço, Promoção, Observações)
R9	<ul style="list-style-type: none"> • eliminar prato
R10	<ul style="list-style-type: none"> • editar dados do restaurante (Logótipo, Localização, Pagamentos, Contactos, Horário, Galeria de fotos)
R11	<ul style="list-style-type: none"> • Ativar o restaurante diariamente com um botão ON/OFF
	O Cliente pode no <i>front-office</i> :
R12	<ul style="list-style-type: none"> • Visualizar alguns pratos em cartaz
R13	<ul style="list-style-type: none"> • Pesquisar pratos por nome ou descrição

ESPECIFICAÇÃO DOS CASOS DE USO:

Use Case	UC1 - Regista-se na Plataforma
Pré-condição	---
Actor Principal	Restaurador Proponente
Descrição	O restaurador proponente (proprietário de restaurante) regista-se na plataforma e o Administrador recebe um email com os dados.

Use Case	UC2 – Faz Login
Pré-condição	Previamente registou-se na plataforma e a conta encontra-se ativa, pelo Administrador
Actor Principal	Restaurador
Descrição	O restaurador autentica-se no sistema.

Use Case	UC3 - Insere, edita, ou elimina um prato
Pré-condição	Estar ativo pelo administrador e ter login feito
Actor Principal	Restaurador
Descrição	O restaurador insere um prato, edita-o e pode elimina-lo.

Use Case	UC4 – Ativa/Desativa diária.
Pré-condição	Estar ativo pelo administrador e ter login feito
Actor Principal	Restaurador
Descrição	O restaurador ativa a diária ou desativa-a.

Use Case	UC5 – Editar dados restaurante
-----------------	--------------------------------

Pré-condição	Estar ativo pelo administrador e ter login feito
Actor Principal	Restaurador
Descrição	O restaurador pode editar todos os seus dados.

Use Case	UC6 – Fazer <i>Logout</i>
Pré-condição	Estar ativo pelo administrador e ter login feito
Actor Principal	Restaurador
Descrição	O restaurador encerra a sua sessão.

Use Case	UC7 – Visualiza pratos em cartaz sugeridos
Pré-condição	Alguns pratos estão ativos com a diária ativa de um restaurador
Actor Principal	Cliente
Descrição	Um cliente visualiza pratos ativos sugeridos aleatoriamente.

Use Case	UC8 – Pesquisa pratos
Pré-condição	Pratos estão ativos com a diária de um restaurador ativo
Actor Principal	Cliente
Descrição	Um cliente pesquisa alguns pratos ativos pelo nome ou descrição.

Use Case	UC9 – Pesquisa restaurante
Pré-condição	Restaurantes têm que estar ativos na plataforma
Actor Principal	Cliente
Descrição	Um cliente pesquisa restaurantes pelo nome ou pela descrição.

Use Case	UC10 – Visualiza proximidades
Pré-condição	Cliente autoriza dar a sua localização
Actor Principal	Cliente
Descrição	Um cliente visualiza restaurantes ativos nas suas proximidades.

Use Case	UC11 – Visualiza restaurante
Pré-condição	Restaurante ativo pelo administrador
Actor Principal	Cliente
Descrição	Um cliente visualiza os dados de um restaurante.

Use Case	UC12 – Visualiza trajeto
Pré-condição	Cliente autorizou dar a sua localização
Actor Principal	Cliente
Descrição	Um cliente visualiza o trajeto até ao restaurante.

Use Case	UC13 – Autentica-se
Pré-condição	---
Actor Principal	Administrador
Descrição	O administrador abra uma sessão.

Use Case	UC14 – Ver dados restauradores
Pré-condição	Sessão está aberta
Actor Principal	Administrador
Descrição	O administrador vê os dados dos restauradores.

Use Case	UC15 – Encerra sessão
Pré-condição	Sessão aberta previamente
Actor Principal	Administrador
Descrição	O administrador encerra a sessão.

Use Case	UC16 – Ativa/desativa restaurador
Pré-condição	Administrador autenticado e restaurador proponente registado inserido
Actor Principal	Administrador
Descrição	O administrador ativa ou desativa um restaurante na plataforma.

Use Case	UC17 – Envia email
Pré-condição	Administrador autenticado e Restaurador proponente registado
Actor Principal	Administrador
Descrição	Um email é enviado ao restaurador quando o seu restaurante é inserido

4.4 MATRIZ DE RASTREABILIDADE REQUISITOS/CASOS DE USO

É aqui apresentada a matriz de rastreabilidade entre casos de uso e requisitos, a qual permite fazer o *trace* de cada caso de uso para o requisito que o motivou.

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13	UC14	UC15	UC16	UC17
R1	X																
R2													X	X			
R3													X	X			X
R4													X	X		X	
R5		X															
R6							X										
R7		X	X														
R8		X	X														
R9		X	X														
R10		X															
R11		X		X													
R12							X										
R13								X									
R14									X								
R15										X							
R16											X						
R17												X					
R18							X	X	X	X	X	X					
R19		X															

4.5 MODELO DE DOMÍNIO

Apuramos o modelo de domínio seguinte:

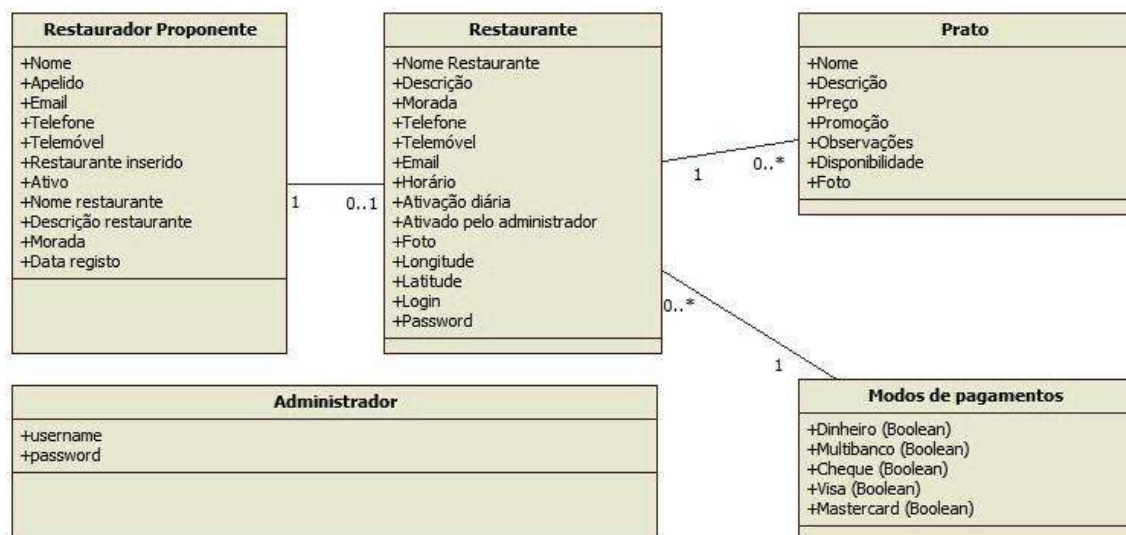


Figura 20: Modelo de Domínio ilustrando Entidades e Relacionamentos.

4.6 NOTAS FINAIS

A especificação dos Casos de Uso e a matriz de rastreabilidade de Casos de Uso/Requisitos serão úteis na parte de teste da aplicação (§7).

Temos agora a solução desenhada para começar com o desenvolvimento do trabalho, é o que vamos descrever no capítulo seguinte.

5. DESENVOLVIMENTO DO PROJETO

5.1 INTRODUÇÃO

O trabalho Pratos Em Cartaz foi desenvolvido apoiando-se numa metodologia de desenvolvimento de tipo ágil, quer dizer com reuniões frequentes (semanalmente ou quinzenalmente) para obter *feedback* das partes envolvidas e selecionar os requisitos a considerar na iteração seguinte do desenvolvimento. Descrevemos abaixo a arquitetura da solução, a base de dados, as tecnologias envolvidas, e a interface da solução.

5.2 ARQUITETURA DA SOLUÇÃO

O sistema de base de dados utilizado é o MySQL. É *Open Source* (licença GPL) e pertence agora à Oracle. É um SGBD – Sistema de Gestão de Base de Dados – relacional. Segundo o próprio site Web (<http://www.mysql.com>) é a «base de dados *Open Source* mais popular em todo o mundo». Tem um excelente desempenho e estabilidade, é fácil de usar e é pouco exigente quanto a recursos de hardware, é portátil suportando qualquer plataforma atualmente [23]. Para ligar à base de dados utilizamos a interface Web da linguagem PHP e uma ferramenta ORM (Redbean <http://redbeanphp.com>).

Para o servidor Web usamos o Apache para poder correr o código PHP do lado do servidor.

O cliente é qualquer dispositivo munido de um navegador Web para servir a aplicação Web. O código *HTML*, *CSS*, *Javascript* é interpretado do lado do cliente.

Na Figura 21 podemos ver o diagrama de *Deployment* da solução desenvolvida:

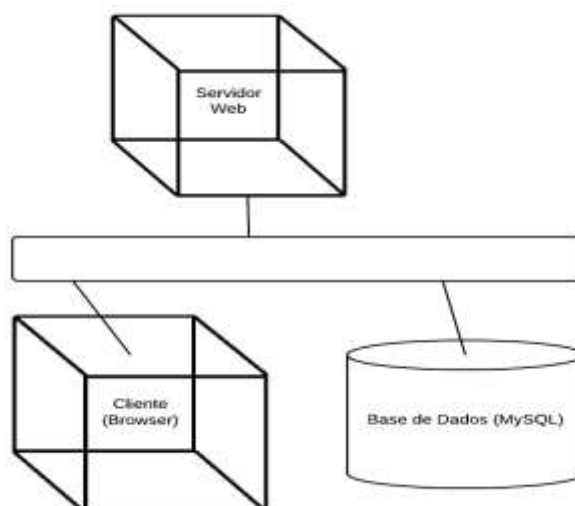


Figura 21: Modelo de Deployment.

Podemos ver o modelo de componentes da solução desenvolvida, na Figura 22.

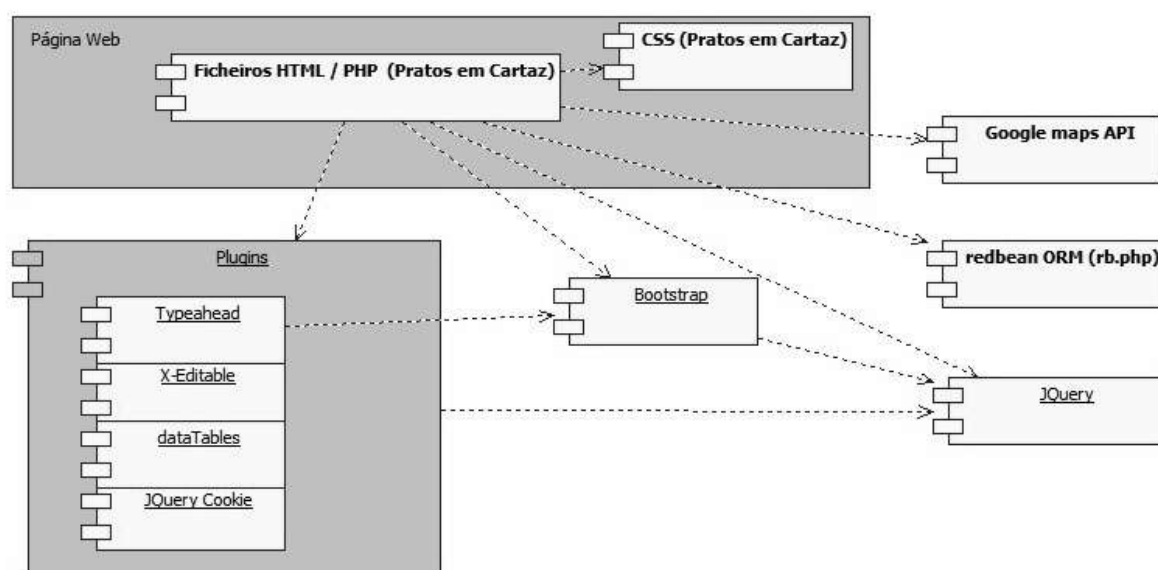


Figura 22: Modelo de Componentes.

Os ficheiros PHP e HTML que compõem a aplicação desenvolvida, juntamente com as suas folhas de estilo, dependem do Bootstrap (analisado na secção 3.7 Bootstrap) e JQuery (cuja função foi explicada na secção

2.4 Tecnologias e conceitos), assim como de diversos plugins que deles também dependem e que são explicados na secção 5.4 Tecnologias envolvidas. Para além disso, é usada a Framework de mapeamento objeto-relacional Redbean, para facilitar o acesso à base de dados, e a API do Google Maps para as funcionalidades envolvendo visualização e navegação em mapa.

5.3 BASE DE DADOS

O esquema da base de dados é apresentado na figura seguinte (Figura 23), através do diagrama Entidades-Relacionamento (modelo E-R), o qual foi obtido localmente a partir do programa MySQL Workbench com a opção no menu Database > Reverse Engineer.

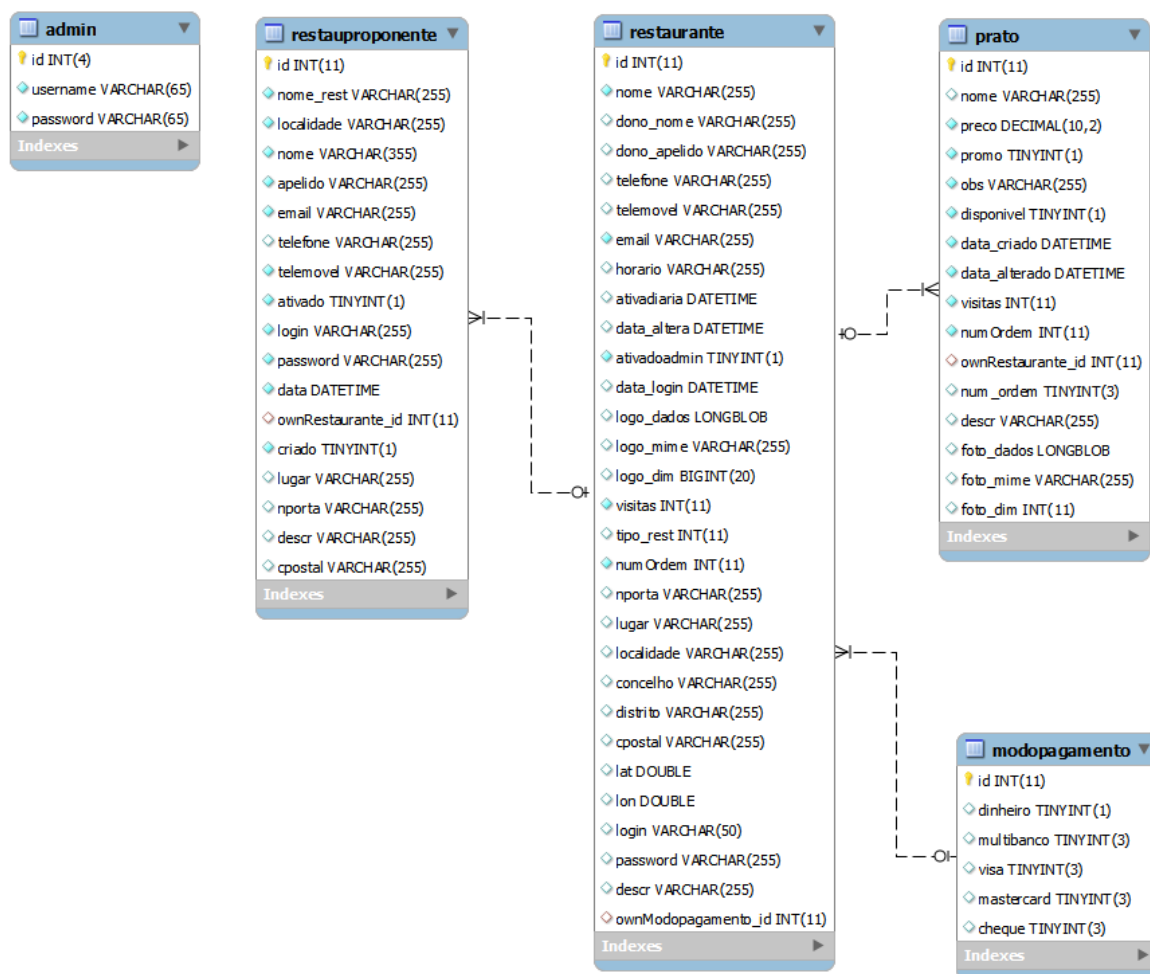


Figura 23: Esquema da Base de Dados.

5.4 TECNOLOGIAS ENVOLVIDAS

Como vimos no Modelo de componentes na secção §5.2 (Figura 22) utilizámos vários *plugins* à volta da biblioteca JQuery. São eles o *x-editable* para disponibilizar a edição de dados diretamente *inline* em campo de texto *input*, *textarea*, *checkbox*, no back-office do restaurador. Outro *plugin* JQuery é o *dataTables* permitindo uma apresentação de tabela formatada com paginação e pronta para a ordenação por ordem alfabética das diferentes colunas, este *plugin* é utilizado no *back-office* do administrador.

Outro *plugin* útil é o *typeahead* que permitiu a apresentação de sugestão de resultados enquanto se escreve numa caixa de texto antes de lançar a pesquisa a um prato ou a um restaurante. Este *plugin* está ligado ao Bootstrap que o disponibiliza. O Bootstrap é constituído por componentes CSS assim como Javascript através do JQuery.

Outro *plugin* útil é o de *JQuery-Cookie* que permite sobretudo gerir a mudança de tema, um de fundo escuro e o outro de fundo claro, de toda a aplicação sem esquecer os *back-offices* administrador e restaurador. Ter um tema escuro ou um fundo branco permite à aplicação ser ágil na resposta à luminosidade envolvida enquanto é usado o dispositivo.

Isto aumenta a sua usabilidade tanto num ambiente interior com pouca luz como num ambiente exterior com muita luz. Neste último caso, por exemplo, poderia ser difícil lidar só com um fundo escuro que refletiria demasiado a luz sem deixar visualizar o conteúdo, o que pode acontecer com frequência, e é assim justificada a utilidade de um fundo mais luminoso da aplicação por razões práticas de comodidade de utilização.

Outro *plugin* de natureza Javascript é aquele ligado à API de Google maps. A maioria das funcionalidades desta API é utilizada como por exemplo apresentação de marcadores e seus arrastamentos para outro lugar, desenhar um círculo no mapa à volta da posição do utilizador, traçar trajetos e alterá-los, a navegação de a posição atual até ao restaurante escolhido, diferentes camadas como a de meteorologia, etc..

Exemplo de código Javascript para desenhar um círculo com a API Google maps:

```
//Draw circle *****
    center = minhapos;
    var circle = new google.maps.Circle({
        center: center,
        map:map,
        radius:15000,
        strokeColor: "#EEEEEE",
        strokeOpacity:0.3,
        strokeWeight: 3,
        fillColor: "#000000",
        fillOpacity:0.1,
        draggable:false,
        editable:false,
        clickable:false,
        geodesic:true
    });
    circle.bindTo('center',mark,'position');
//End draw Circle *****
```

As páginas Web estão codificadas com HTML5 disponibilizando funcionalidades inovadoras. Muitos recursos do HTML5 têm sido construídos com a consideração de ser capaz de executar em dispositivos de baixa potência como *smartphones* e *tablets* [24]. Esta nova tecnologia permite por exemplo oferecer ao utilizador através do *browser* a possibilidade de dar a sua posição geográfica.

Exemplo de código para dar a geolocalização do navegador:

```
<script type="text/javascript">
    // Try HTML5 geolocation
    var mylon; var mylat;
    function geolocalizame() {
        if(navigator.geolocation) {

            navigator.geolocation.getCurrentPosition(function(position) {
                var minpos = new google.maps.LatLng(
                    position.coords.latitude,
                    position.coords.longitude);

                mylat = position.coords.latitude;
                mylon = position.coords.longitude;
                map.setCenter(minpos);
            }, function(error) {
                alert(error.message + '\n' +
                    'Erro ao dar a sua localização');
            }
            , { //hack for some Android devices
                enableHighAccuracy: true ,timeout : 5000
            });
        } else {
            // Browser doesn't support Geolocation
            alert ("O Seu Navegador não suporta Geolocalização");
        }
    }

    //END try HTML5 geolocation
</script>
```

O HTML5 permite também usar o código seguinte para lançar a aplicação de telefonia:

```
<a href="tel:<?=$rest->telefone;?>"><?=$rest->telefone;?>"></a>
```

Para gerir as consultas à base de dados utilizamos uma ferramenta de ORM nomeada *redbean* (www.redbeanphp.com). Esta ferramenta está disponível nos termos de duas licenças Open Source, New BSD e GPLv2. Esta *Library* é útil na medida que simplifica o

código utilizado mapeando objetos para e de a base de dados, e assim permite facilitar e acelerar o desenvolvimento de uma aplicação Web. É simples de implementar bastando basicamente incluir a biblioteca *redbean* na página PHP:

```
<?php require '../library/rb.php'; ?>
```

As imagens transferidas (*upload*) pelo utilizador restaurador estão armazenadas na base de dados MySQL com o tipo Blob (*binary large object*). A apresentação destas imagens (pratos e restaurante) é feita com um script utilizando o ID da imagem na tabela relacionada.

Exemplo de linha de código para apresentar a imagem de um prato:

```
nome?>" class="prato img-circle">
```

O script *vista_prato.php* contém por exemplo as linhas seguintes:

```
header('Content-type: ' . $prato['foto_mime']); //jpg gif png
header('Content-length: ' . $prato['foto_dim']); //tamanho em KB
echo $prato['foto_dados']; //Dados numéricos da imagem (tipo
Blob na BD)
```

Terminamos aqui as considerações sobre as tecnologias envolvidas e vamos ilustrar, no capítulo seguinte, a interface do utilizador, recorrendo a imagens da interface da própria aplicação.

5.5 INTERFACE DA SOLUÇÃO

Nesta secção, vamos ilustrar a interface do utilizador recorrendo a imagens da interface da própria aplicação. Utilizámos o programa *Android Screen Monitor* (<https://code.google.com/p/android-screen-monitor/>) para projetar o ecrã do dispositivo Android para o ecrã do PC e fazer cópias de ecrã.

As Figuras 24, 25 e 26, ilustram a interface da aplicação, no Android, para um utilizador anónimo. A interface tem 3 grupos de funcionalidades no *layout* aqui apresentado em 3 grupos respetivamente à disposição vertical: o acolhimento (Figura 24), a visualização de sugestão de pratos (Figura 25) e por último a pesquisa (Figura 26).

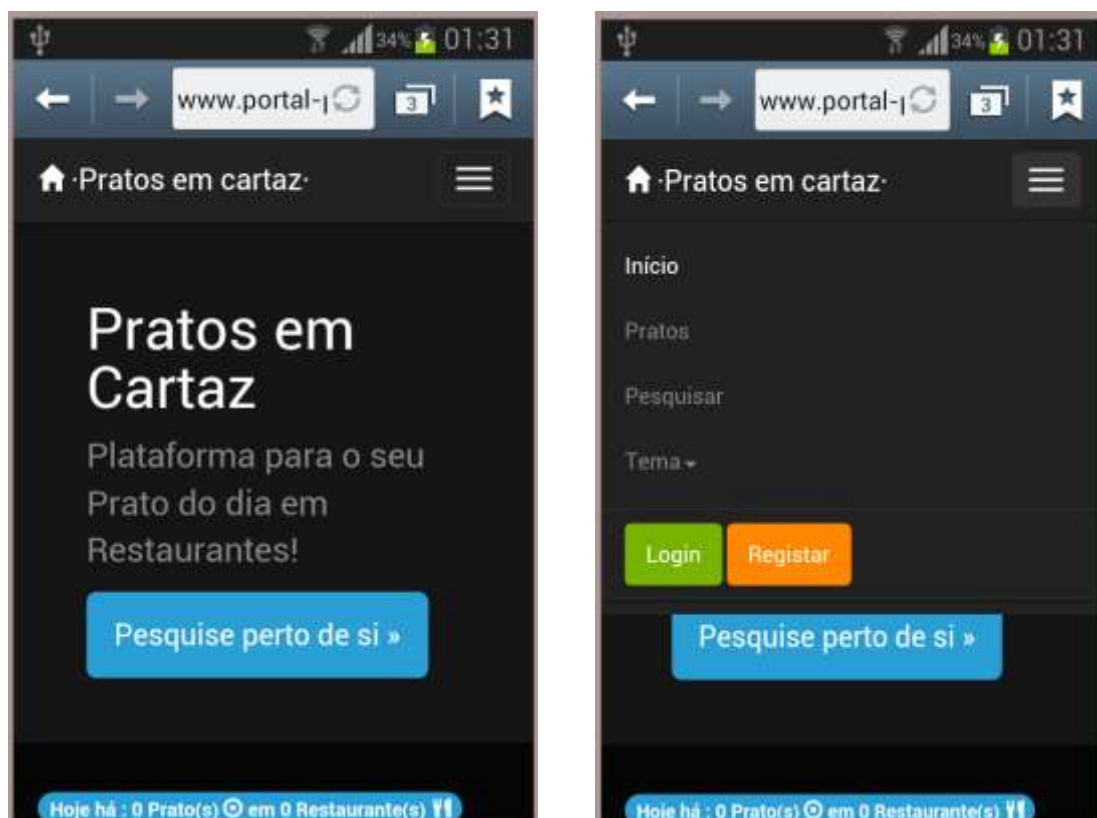


Figura 24: Ecrãs de acolhimento no Android (à direita com o menu aberto)

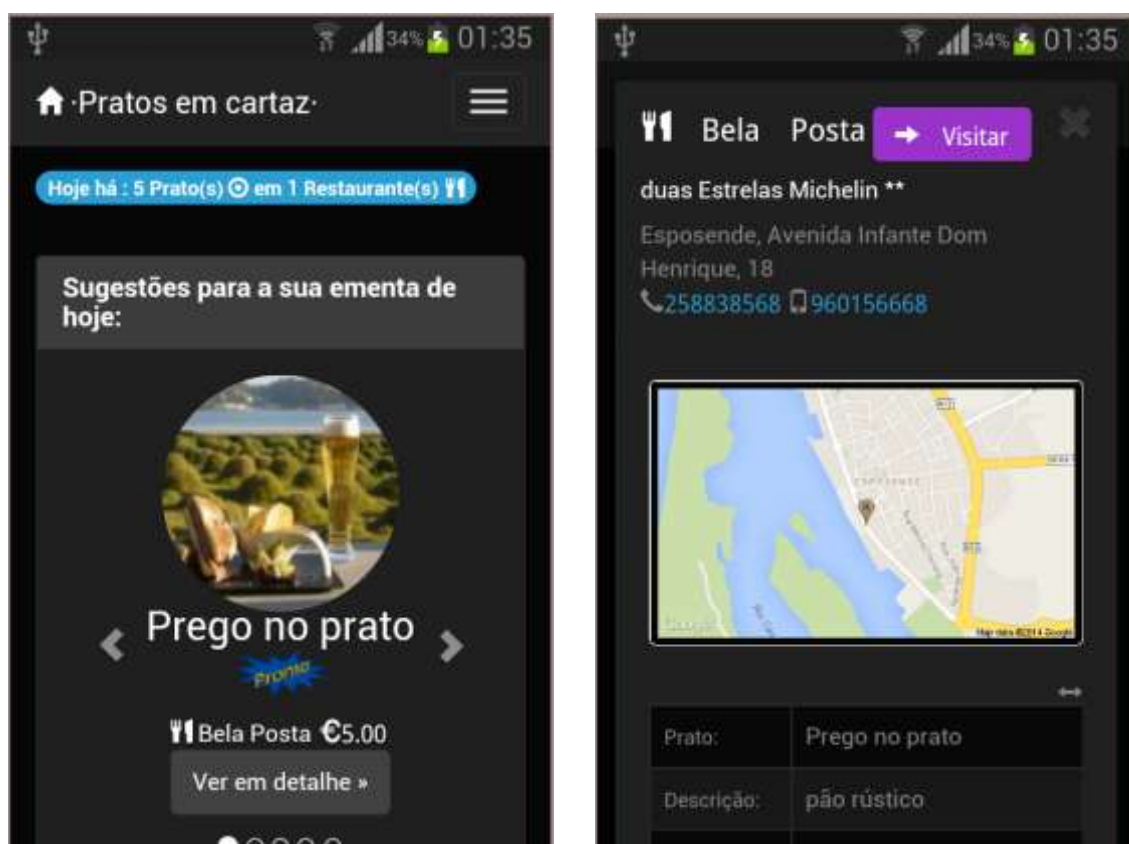


Figura 25: Interface de visualização de sugestão de pratos e seu detalhe.

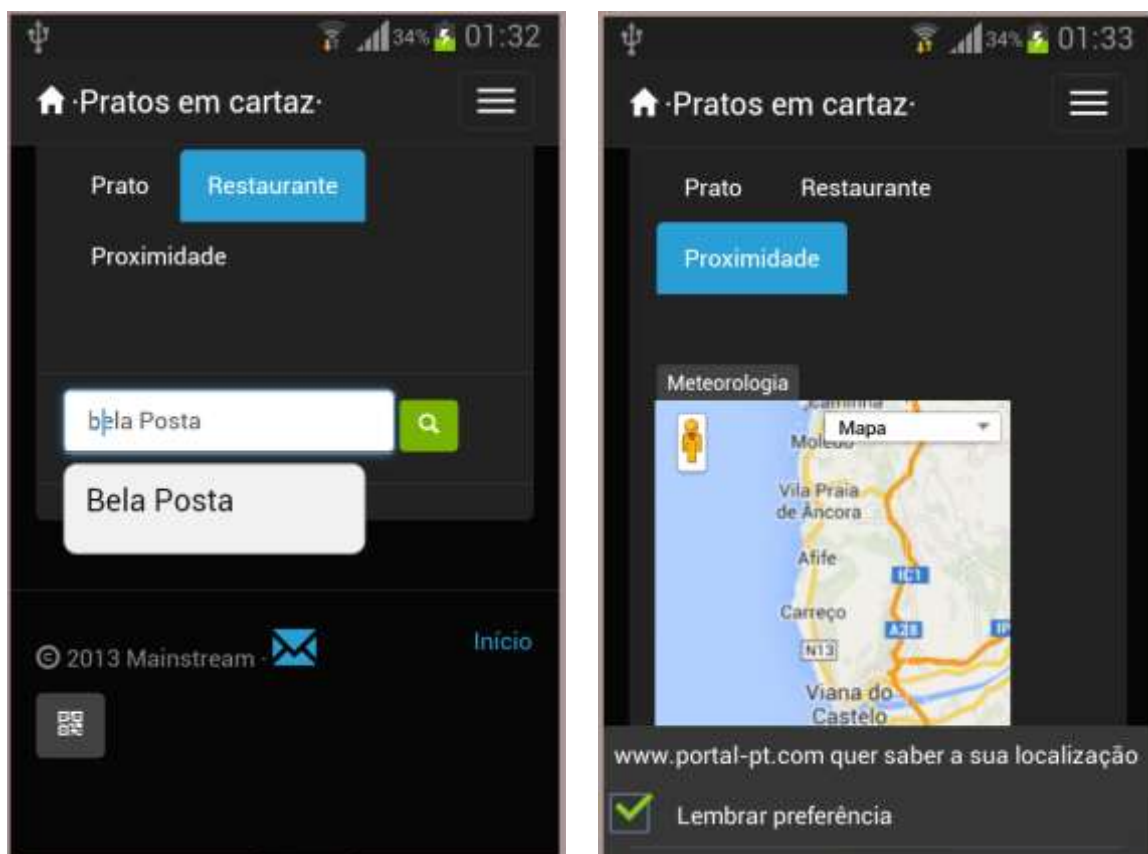


Figura 26: Interface de pesquisa de restaurantes (à esquerda) e de proximidade (à direita).

As figuras seguintes mostram, agora, a interface de a página de um restaurante. É constituída por 3 níveis: a primeira parte, não ilustrada, contém a foto do restaurante, nome e dados. A segunda parte apresenta as formas de pagamento e horário de abertura do restaurante (lado esquerdo da Figura 27). A terceira parte apresenta a ementa fechada para não ocupar demasiado espaço se os pratos forem muitos (lado direito da Figura 27). A quarta parte mostra a ementa aberta (lado esquerdo da Figura 28) e passos para a localização do restaurante (lado direito da Figura 28). O utilizador pode dar a sua localização e traçar o trajeto até ao restaurante com as opções “de carro” ou “a pé”. É apresentado o caminho com todos os passos da API Google *maps*. Há também possibilidade de “iniciar a navegação” na aplicação GPS Google Maps do Android com orientação áudio para se deslocar até ao restaurante escolhido.

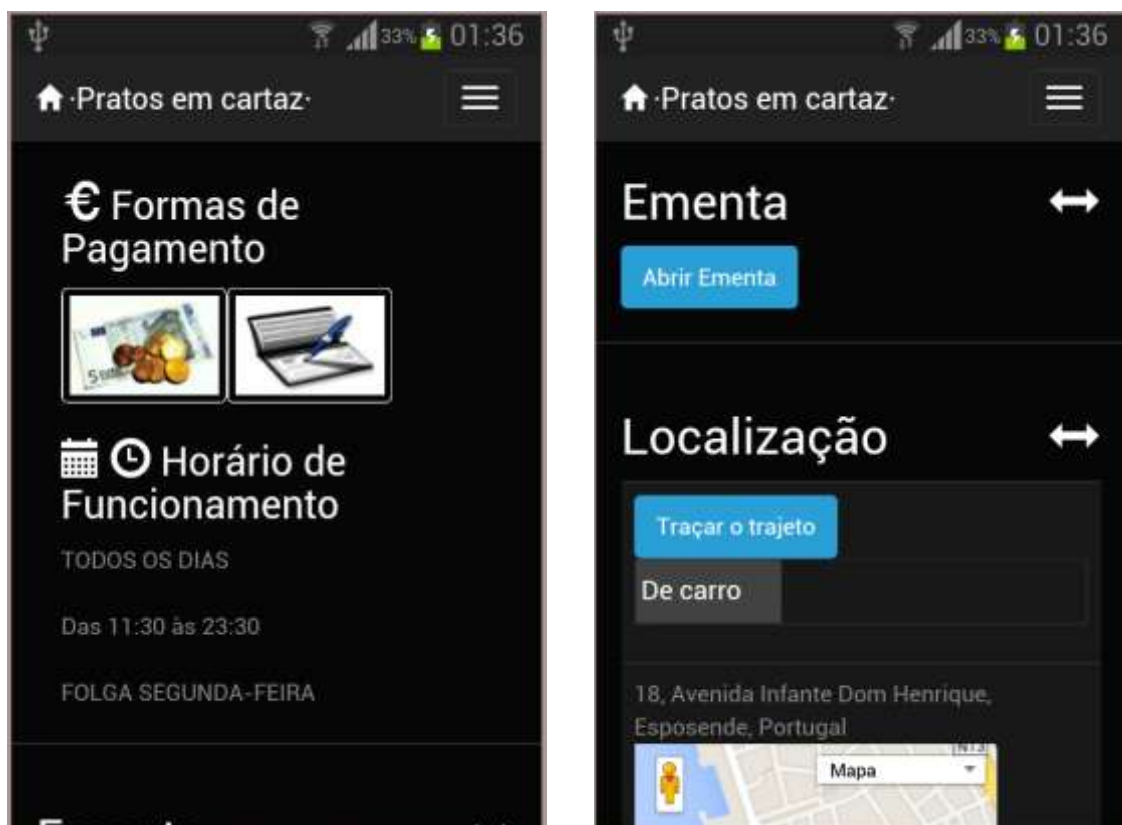


Figura 27: Interface de visualização de um restaurante.
(Formas de pagamento e horário à esquerda, ementa fechada e localização à direita)

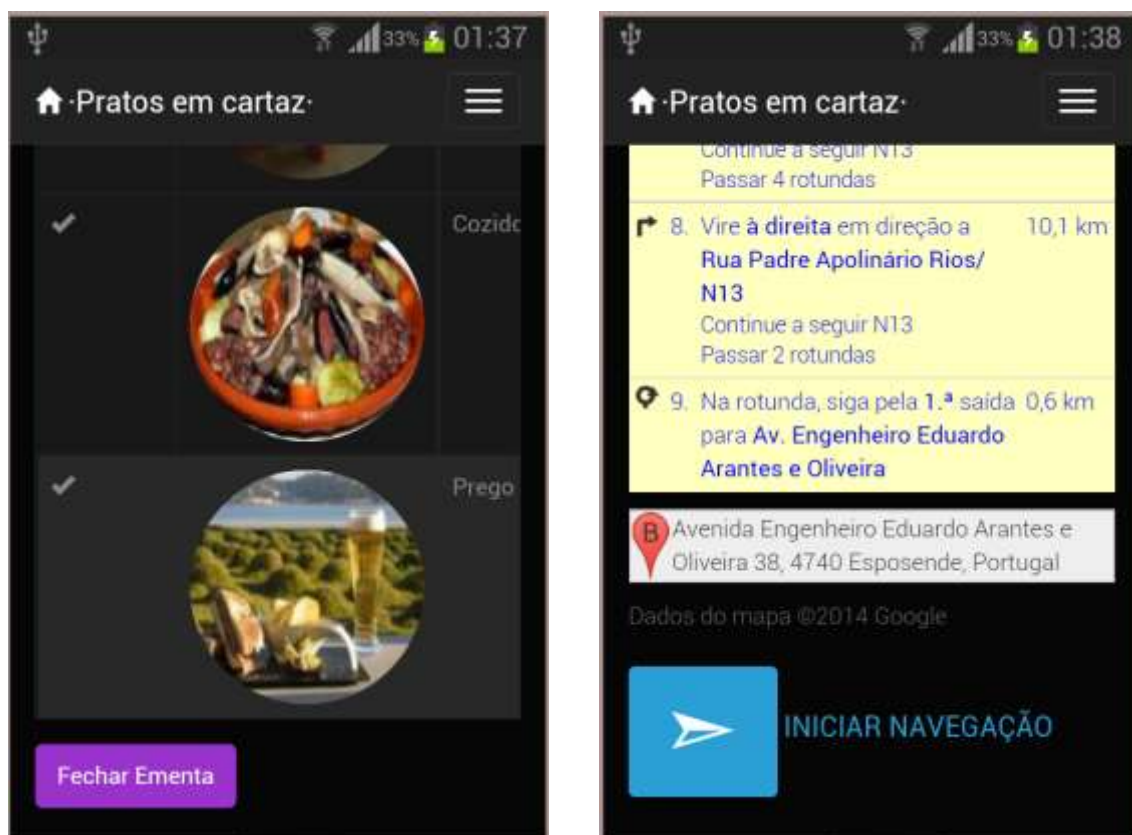


Figura 28: Interface de visualização de um restaurante
Ementa aberta à esquerda e passos para a localização do restaurante à direita

Quanto à interface do *back-office* do restaurador, a Figura 29 ilustra os ecrãs de edição da localização (determinada por geolocalização), e dos restantes campos do restaurador e restaurante.

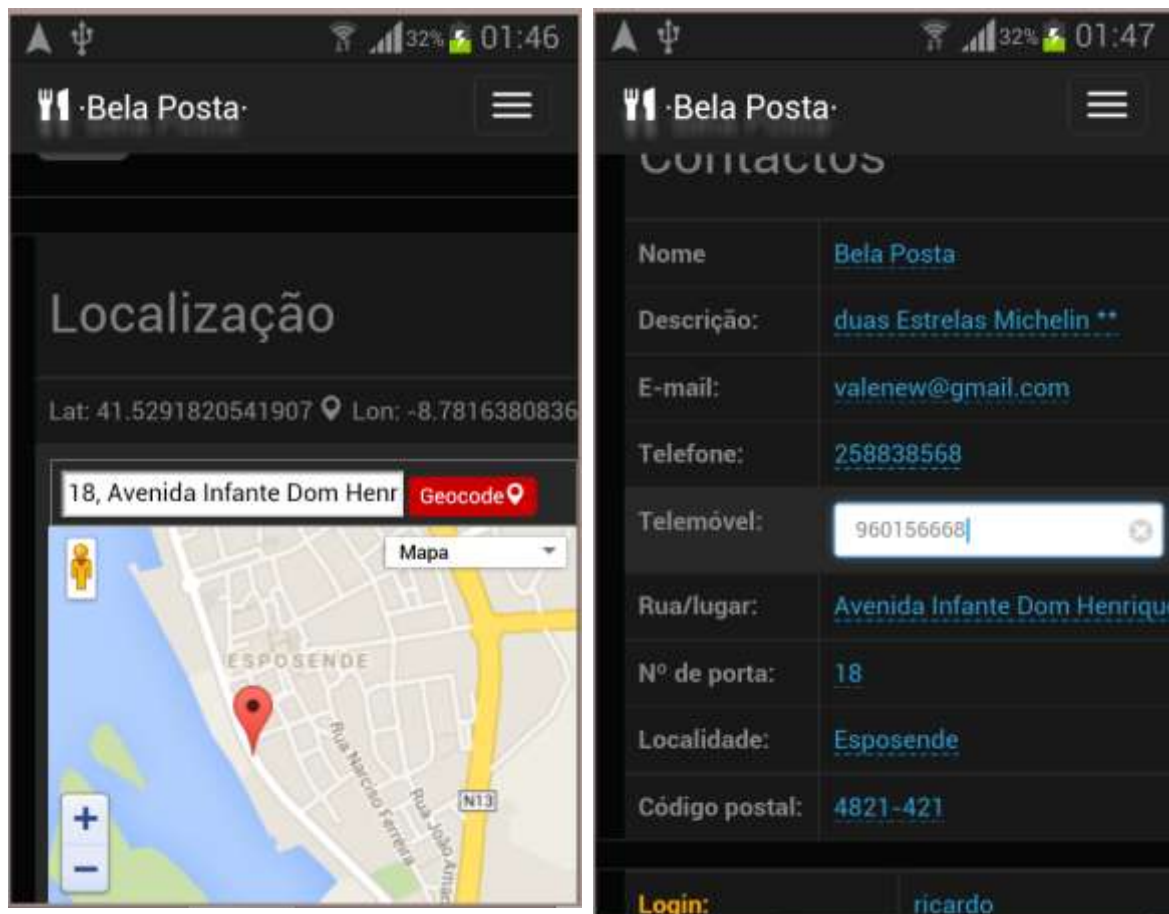


Figura 29: Interface do *back-office* do restaurador para editar localização e os dados

Quanto à interface do *back-office* do administrador, a Figura 30 ilustra o ecrã principal que permite inserir um restaurante proponente na plataforma Pratos em Cartaz, ou apagá-lo. E permite também ativar e desativar um restaurante da plataforma.

Os dados aparecem nesta interface de administração posteriormente a um restaurador proponente ter preenchido o formulário de registo com todos os dados pedidos obrigatoriamente.

Administração | Praia em Callaz
Toma

Vista
Logout

Restauradores Proponentes

Mostra 15
Linhas por página

	Nome do Restaurante no sistema	Descrição	Localidade	Cód. Postal	Lugar	Nº	Nome	Apelido	E-mail	Telefone	Telemóvel	Data	Ativado	
1	Muralha da China	Restaurante Chinês	Viana do Castelo	4900-000	Av. do Novo Gê Eanes	33	Ann	Xiang	xiang@gmail.com	258888888	960156668	2014-03-15 18:07:37	<input checked="" type="checkbox"/>	<input type="button" value="Apagar"/>
2	Os três Poles	no centro histórico	Viana do Castelo	4900	Praça da República	18	Teresa	Silva	valenciano@sapo.pt	258835068	960156668	2013-12-28 21:42:03	<input checked="" type="checkbox"/>	
3	Areia	Freixo ao mar	Carreço	4900-270	Praça de Carreço	0	José	Viana	valenevi@gmail.com	258835068	960156668	2013-12-18 15:17:00	<input checked="" type="checkbox"/>	
4	Scala	Restaurante grande, frente ao mar	Viana do Castelo	4900-341	Avenida do Abadeiro	1	José Manuel	Scala	zeta@scalars.pt	919293946	919293940	2013-12-18 15:48:27	<input checked="" type="checkbox"/>	
5	Bora Praia		Vila do Conde				Ricardo	Vale	valenevi@gmail.com	258731350	960156668	2013-11-27 11:40:34	<input checked="" type="checkbox"/>	

Mostrando 1 a 5 de 5 entradas

1

Figura 30 Interface do *back-office* do Administrador para inserir/apagar e ativar/desativar restaurantes

5.6 NOTAS FINAIS

O projeto consiste numa aplicação Web responsiva, a qual se consegue adaptar a diferentes tamanhos de ecrã. As imagens ilustradas neste capítulo são do dispositivo de teste, um ecrã de *smartphone* Samsung Galaxy SII GT-I9100, com Android Jelly Bean 4.1.2. Foi massivamente usada a tecnologia Ajax para tornar a interface reduzida numa única página com a vantagem de não carregar cada vez uma página Web nova à cada ação realizada.

A interface obtida sobretudo graças ao *framework* Twitter Bootstrap tem um aspeto atrativo na minha opinião, como no caso de aplicações móveis nativas bem construídas. A vantagem aqui é que pode ser visualizado em qualquer dispositivo móvel ou não-móvel, e funcionando com qualquer sistema operativo equipado com um *browser*.

O desenvolvimento faz-se bem com as tecnologias escolhidas à volta da Web como por exemplo *HTML5*, *Javascript*, *Jquery*, *CSS*, *PHP*, *MySQL* etc. Estas tecnologias parecem adequadas para aprofundar e levar a bem o projeto proposto pela empresa *Mainstream*. O *feedback* recebido da empresa é de que tudo funciona perfeitamente. É o que vamos testar no capítulo seguinte.

6. AVALIAÇÃO OU DISCUSSÃO DOS RESULTADOS DO TRABALHO

6.1 INTRODUÇÃO

Este capítulo apresenta casos de testes para validação do sistema face aos requisitos definidos inicialmente. É também feita uma discussão sobre a usabilidade do sistema, em usando como referência as heurísticas de Nielsen. Veremos também a otimização do carregamento da aplicação através do *Google insight Pagespeed*, e para finalizar é relatada a utilização da ferramenta *Loadstorm* para os testes de carga e performance.

6.2 TESTES DE VALIDAÇÃO

Esta seção é dedicada à validação do sistema face aos requisitos definidos. A validação é feita por alguns Casos de Teste sendo apresentado, para cada caso de teste, o seu Objetivo (o que é testado), Passos do teste, Resultado esperado e Resultado verificado. Para cada caso de teste são ainda apresentados os requisitos que se pretende validar no sistema, permitindo assim a rastreabilidade entre requisitos (apresentados na secção 4.2 Requisitos do projeto (com enunciado)) e casos de teste.

CASO DE TESTE 1

Requisito Testado	Objetivo	Passos do Teste	Resultado Esperado	Resultado Verificado
R1	O Restaurador Proponente regista-se no site Web.	O restaurador acede ao URL de registo e introduz os seus dados. Submete o formulário ao clicar no botão	O administrador da plataforma deve receber um email com os dados introduzidos pelo restaurador proponente e estes dados devem aparecer também no <i>back-office</i> do administrador.	O administrador recebe os dados e clica num link para ir ao seu <i>back-office</i> e depois de autenticar-se verifica que estão lá os dados todos numa tabela. O restaurador proponente também recebe um mail para ficar a aguardar.

CASO DE TESTE 2

Requisito Testado	Objetivo	Passos do Teste	Resultado Esperado	Resultado Verificado
R1	Ao registrar o Restaurador tem que entrar todos os seus dados corretamente antes de poder submeter o formulário.	Um campo não é preenchido ou não respeita o formato pedido.	Uma mensagem de erro clara aparece ao utilizador para poder corrigir e o formulário não é submetido.	Conforme esperado

CASO DE TESTE 3

Requisito Testado	Objetivo	Passos do Teste	Resultado Esperado	Resultado Verificado
R2 R3	O Administrador valida a inscrição do restaurador proponente	O administrador autentica-se e clica no botão para inserir o restaurante na plataforma.	Quando o Administrador valida a inscrição, o sistema envia um email para dizer qual o Login e Password disponibilizado para o restaurador proponente e insere na BD os dados para o novo restaurante.	Conforme esperado

CASO DE TESTE 4

Requisito Testado	Objetivo	Passos do Teste	Resultado Esperado	Resultado Verificado
R2 R3	O Administrador rejeita a inscrição do restaurador proponente	O administrador autentica-se e clica no botão para rejeitar o restaurante na plataforma.	Quando o Administrador rejeita o restaurador/restaurante, ele é apagado do back-office e da BD.	Conforme esperado

CASO DE TESTE 5

Requisito Testado	Objetivo	Passos do Teste	Resultado Esperado	Resultado Verificado
R4	O Administrador procede à ativação da conta do restaurador e restaurante.	O administrador autentica-se e clica no botão para ativar um restaurador/restaurante na plataforma. Após isso, o restaurador com conta ativada autentica-se no sistema. Por fim, um utilizador anónimo pesquisa pelo restaurante ativado, e visualiza os seus dados, pratos e localização no mapa.	Como a conta de restaurador está ativa, as ocorrências e funcionalidades deste aparecem normalmente.	Conforme esperado.
R14				

CASO DE TESTE 6

Requisito Testado	Objetivo	Passos do Teste	Resultado Esperado	Resultado Verificado
R4	O Administrador procede à desativação da conta do restaurador e restaurante.	O administrador autentica-se e clica no botão para desativar um restaurador/restaurante na plataforma. Após isso, o restaurador	Com a conta desativada pelo Administrador, o restaurador não consegue autenticar-se e o restaurante e seus produtos não aparecem na plataforma em lado nenhum. Se o restaurador tenta fazer login com a conta desativada, aparece-lhe a mensagem: "A sua conta	Conforme esperado.

		com conta ativada tenta autenticar-se no sistema. Por fim, um utilizador anónimo pesquisa pelo restaurante desativado.	foi desativada” Um restaurante desativado não aparece na pesquisa, seus produtos também não, e se tentarmos aceder ao URL dedicado ao restaurante é apresentada a mensagem seguinte: “O restaurante não está ativo...”.	
--	--	---	--	--

CASO DE TESTE 7

Requisito Testado	Objetivo	Passos do Teste	Resultado Esperado	Resultado Verificado
R5 R6 R19	O restaurador pode autenticar-se e encerrar uma sessão.	O restaurador acede à página de autenticação e entra as suas credenciais, clique no botão Login e entra no seu <i>back-office</i> se as suas credenciais foram corretas e existiram. Pode também encerrar a sessão.	O restaurador clique no botão <i>Login</i> da página principal para aceder à página de autenticação e entra corretamente as suas credências. É-lhe permitida entrada para o seu <i>back-office</i> , porque já foi registado na plataforma e sua conta está ativa. O restaurador tendo aberto uma sessão pode encerrar esta clicando o botão de <i>Logout</i> situado na parte superior da página na barra de navegação.	Conforme esperado.

CASO DE TESTE 8

Requisito Testado	Objetivo	Passos do Teste	Resultado Esperado	Resultado Verificado
R7 R8	O restaurador pode depois de autenticar-se inserir	O restaurador autentica-se,	Os novos pratos são inseridos na BD.	Conforme esperado.

R9	um prato, editar um prato (com os campos: Disponibilidade, Descrição, Preço, Promoção, Observações) e pode também eliminar um prato.	<p>e clique no botão com um signo “+” para entrar um prato novo na sua ementa e preenchendo os vários campos de seguida.</p> <p>O restaurador pode editar estes dados de prato clicando em cima do texto que é editável diretamente e depois validando ou anulando a modificação.</p> <p>O restaurador elimina um prato clicando o botão “X” com fundo vermelho à direita do prato.</p>	<p>Todos os dados dos pratos são editados e apresentados no <i>backoffice ao restaurador</i>.</p> <p>Os pratos são eliminados quando solicitado pelo restaurador.</p>	
----	--	---	---	--

CASO DE TESTE 9

Requisito Testado	Objetivo	Passos do Teste	Resultado Esperado	Resultado Verificado
R10	Editar dados do restaurante (Foto, Pagamentos, Horário, Diária, Contactos, Localização, Credenciais).	O restaurador autentica-se e modifica todos os seus dados disponíveis.	Todos os dados são editados para a BD e apresentados no <i>back-office</i> ao restaurador.	Conforme esperado.

CASO DE TESTE 10

Requisito Testado	Objetivo	Passos do Teste	Resultado Esperado	Resultado Verificado
R11	Ativar o restaurante diariamente.	O restaurador autentica-se e ativa o seu restaurante para os seus pratos aparecerem nas pesquisas do dia.	O restaurante com a ativação diária efetuada, tem os seus pratos a aparecem nas pesquisas efetuadas.	Conforme esperado.

CASO DE TESTE 11

Requisito Testado	Objetivo	Passos do Teste	Resultado Esperado	Resultado Verificado
R18 R12 R13	Um cliente/utilizador anónimo: Visualiza alguns pratos em cartaz; Pesquisa pratos por nome ou descrição.	Um utilizador anónimo ou cliente acede à página principal e visualiza pratos. O utilizador pesquisa pelo nome ou descrição de um prato.	Visualizar pratos. Visualizar pratos resultado da pesquisa.	Conforme esperado.

CASO DE TESTE 12

Requisito Testado	Objetivo	Passos do Teste	Resultado Esperado	Resultado Verificado
R14 R16 R17	Um cliente pesquisa restaurantes por nome ou descrição. Escolher um restaurante.	Um utilizador anónimo ou cliente acede à página principal e	Pesquisar restaurantes, escolher um restaurante e visualizar o trajeto até à localização do restaurante.	Conforme esperado.

	Visualizar o trajeto até ao restaurante a partir de sua localização	pesquisa restaurantes pelo nome ou descrição. O utilizador escolhe um restaurante. O trajeto até ao restaurante é apresentado.		
--	---	--	--	--

CASO DE TESTE 13

Requisito Testado	Objetivo	Passos do Teste	Resultado Esperado	Resultado Verificado
R15	Um cliente pesquisa restaurantes por localização próxima de si.	Um utilizador anónimo ou cliente acede à página principal e pesquisa restaurantes perto de si.	Visualizar restaurantes em redor da sua localização, escolher um restaurante e visualizar o trajeto até à localização do restaurante.	Conforme esperado.
R16	Escolher um restaurante.	O utilizador escolhe um restaurante.		
R17	Visualizar o trajeto até ao restaurante a partir de sua localização	O trajeto até ao restaurante é apresentado.		

6.3 TESTES DE USABILIDADE / HEURÍSTICAS DE NIELSEN

Vamos agora descrever a interface do ponto de vista da experiência do utilizador. Os 10 pontos seguintes que utilizámos para discutir são retirados das heurísticas de Nielsen que são a referência em matéria de usabilidade [25].

1. Visibilidade de Status do Sistema:

Por exemplo quando um restaurador faz *upload* de uma imagem, ele pode ver uma imagem de ilustração da espera que é um círculo girando para informar que o sistema

está a trabalhar. É útil quando uma imagem pesada (i.e superior a 100 KB) é carregada para o servidor e o *upload* demora mais de 1 segundo, servindo para informar sobre o estado do sistema que está a trabalhar.

2. Relacionamento entre a interface do sistema e o mundo real:

O exemplo de relacionamento com um mapa geográfico de papel impõe-se aqui e é facilmente reconhecível na interface de *Google maps*.

3. Liberdade e controlo do utilizador:

Por exemplo o utilizador está sempre livre de usar a interface enquanto o sistema trabalha em segundo plano com Javascript Assíncrono (Ajax) e a interface não é bloqueada permitindo ao utilizador navegar e carregar em *links* ou botões. A interface permite desistir de uma ação premindo outro botão, como por exemplo no caso de editar dados no *back-office* do restaurador. As janelas modais abrem e fecham em alguns segundos sem obstruir a interface quando desnecessário. O utilizador nunca é envolvido em longas esperas ou longos procedimentos e pode sempre fazer outra coisa sempre que o deseja e desistir a qualquer momento de uma ação.

4. Consistência e standards:

Por exemplo o design e o modo de utilização da interface é sempre o mesmo de uma página para outra. É feito uso de botões similares e uso de cores como o vermelho para indicar ações críticas. As palavras usadas na plataforma são simples e livres de ambiguidade.

5. Prevenção de erros:

Por exemplo na interface do restaurador, é permitido a entrada de valores numéricos com um ponto em vez de vírgula, e isto é corrigido automaticamente. Se uma *string* de caracteres é entrada em vez de um valor numérico este passa a ter o valor zero. O utilizador pode sempre corrigir qualquer valor e cadeia de caracteres logo de seguida clicando simplesmente o campo de caracteres que é sublinhado para sugerir que é um link editável. Outro exemplo é a possibilidade já descrita no ponto 3 onde um trajeto é modificável se achar que é errado ir pelo aquele caminho no mapa geográfico. Outra prevenção de erro é a possibilidade no mapa de alterar o ponto de partida e chegada para o cliente. O restaurador pode também facilmente alterar a localização de seu restaurante se estiver errada à partida a partir do endereço dado em texto, bastando arrastar o marcador do mapa para a localização certa.

6. Reconhecimento ao contrário de lembrança:

O utilizador reconhece uma interface Web com botões e formulários simples de usar como por exemplo um mapa geográfico para indicar um lugar. É usado o *Google maps* que é muito intuitivo e reconhecido como muito popular na Web.

7. Flexibilidade e eficiência de uso

Com um simples clique um utilizador anónimo é capaz de dar a sua localização e traçar o trajeto até a localização do restaurante atualmente visitado naquele momento na Web. Com um simples clique num número de contacto de telefone ou telemóvel é lançado de imediato a aplicação para telefonar bastando premir um único botão para fazer a chamada.

8. Estética e design minimalista:

Toda a aplicação é minimalista e é concentrada em poucas páginas Web de design similar, sendo a maioria das ações realizadas com Ajax sem recarregar de cada vez a página Web. A interface de acolhimento é focada num conteúdo reduzido (i.e um botão para saltar direto à pesquisa em fundo de página, e só um máximo de 10 pratos sugeridos num *slide* de um único elemento de cada vez, estando os outros pesquisáveis). A interface de um restaurante para o cliente só tem 4 secções (Contactos, Pagamentos/Horários, Ementa que está recolhida, e a parte de localização). Portanto é tudo focado naquilo que é importante com um design minimalista e estético graças à *framework Bootstrap*.

9. Ajude os utilizadores a reconhecer, diagnosticar e corrigir erros

Um utilizador visualiza de imediato se um dado foi entrado com erro e pode simplesmente corrigir o erro clicando onde entrou o erro, ou carregando de novo num botão para submeter de novo um formulário. O utilizador quando o deseja pode alterar os dados do mapa alterando por exemplo um trajeto ou ponto de partida e chegada, bastando interagir com o mapa e arrastar marcadores de partida e chegada e também arrastando a linha do trajeto para outra estrada no mapa.

10. Ajuda e documentação

A aplicação é considerada tão simples de utilizar que não foi elaborado nenhuma documentação de ajuda ao utilizador. A interface é muito intuitiva e reduzida focando-se naquilo que importa mais, isto significa ver ou pesquisar um prato, o restaurante associado com os seus contactos e como ir lá. Para o *back-office* do restaurador é também muito simples e intuitivo editar os dados que estão lá. Numa fase futura do projeto poderá ser considerada, se se justificar, uma página com algum texto de ajuda.

6.4 OTIMIZAÇÃO DO CARREGAMENTO DA APLICAÇÃO

Existe uma ferramenta interessante para ajudar-nos a melhorar o desempenho da aplicação Web, é o Google *Insight Pagespeed*, e pode encontrar-se no link seguinte: <http://developers.google.com/speed/pagespeed/insights/>

O link de referência para os testes sobre a nossa aplicação encontra-se em:

http://developers.google.com/speed/pagespeed/insights/?url=www.portal-pt.com%2Fpratos_em_cartaz

Ao início deparei com alguns alertas a vermelho para melhorar. A performance em velocidade melhorou bastante com certos passos executados. O teste passou de 57/100 (alerta vermelha) - no separador Carregamento Telemóveis - para 86/100 (em verde). O separador Computador também está com este resultado aceitável a verde, agora de 86/100.

O que foi feito segundo as sugestões do Google *Insight*:

- removi uma linha do ficheiro bootstrap.css (é o CSS do tema escuro onde estava uma linha de importação de uma fonte de caracteres desnecessária):
`@import url("//fonts.googleapis.com/css?family=Droid+Sans:400,700");`
- Coloquei o código do ficheiro carousel.css dentro do *index.php* (linhas 22 à 97)



Figura 31: Google *Developers PageSpeed Insights* (Análise de carregamento de página Web).

<http://developers.google.com/speed/pagespeed/insights/?url=www.portal-pt.com%2Fpratos-em-cartaz>

O resultado otimizado é aceitável agora, depois de melhoramentos, e vamos passar a executar mais testes de performance na secção seguinte.

6.5 TESTES DE CARGA/PERFORMANCE

Começámos por utilizar a ferramenta *Loadstorm* (www.loadstorm.com) que permite criar um cenário ou script, com vários passos, e fazer um caso de teste de stress a partir deste cenário. O cenário criado envolve os passos de abertura da página principal da nossa aplicação, indo de seguida para a página de login, submetendo um login e

password no formulário de autenticação, para depois clicar num botão. De facto são os passos comuns para um restaurador ativar a sua diária. Depois de correr automaticamente este caso de teste durante 20 minutos com utilizadores indo de 5 a 25 em número simultâneo crescente, obtivemos na parte de análise os 2 gráficos seguintes (Figura 32 e Figura 33):

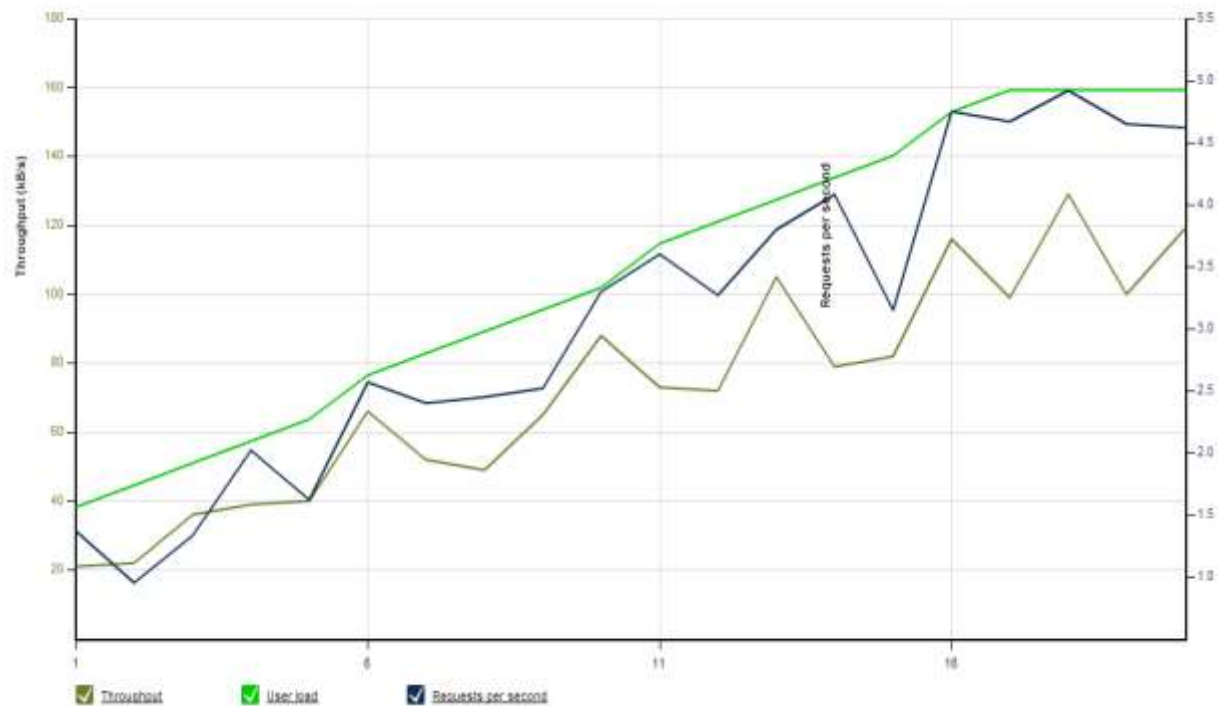


Figura 32: Análise de teste de stress de um cenário com *Loadstorm* para a ativação de diária no Pratos em Cartaz.

Fonte: http://app.loadstorm.com/load_tests/169753 (Link privado)

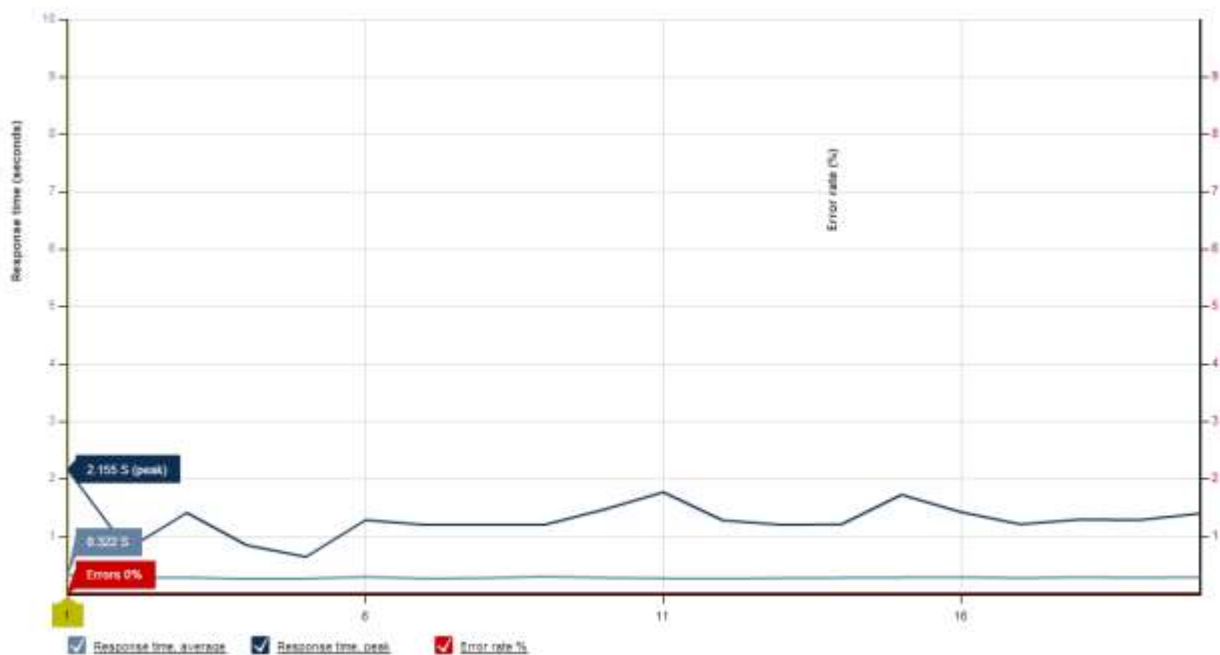


Figura 33: Análise de teste de stress de cenário com *Loadstorm* para a ativação de diária no Pratos em Cartaz.

Fonte: http://app.loadstorm.com/load_tests/169753 (Link privado)

Observámos que o Tempo de Resposta está sempre abaixo de um pico de 2.155 segundos no início de teste. A análise demonstra que o tempo de resposta é bastante bom para o servidor utilizado.

Continuando, consideramos agora outra ferramenta, *Pingdom Tools*. Um caso de teste com o *Pingdom Tools* indica que a aplicação Web carrega em menos de 2 segundos (1,25s neste caso de teste em particular com um total de 5 pratos aparecendo no carroussel da página inicial). O carregamento é feito a partir de Amsterdam, NL que configurámos nos *Settings*. Esta ferramenta é bastante popular na Web e é considerado como referência por alguns Webmasters. É proposto um serviço pago mas também grátis neste link: <http://tools.pingdom.com>. Aqui em baixo encontra-se uma cópia de ecrã ilustrando o nosso caso.



Figura 34: Teste de carga com o *Pingdom Tools*

Fonte: http://tools.pingdom.com/fpt/#!/eDwq2X/www.portal-pt.com/pratos_em_cartaz

6.6 NOTAS FINAIS

Os casos de teste validaram a boa funcionalidade da aplicação. Foram testados todos os requisitos com sucesso.

A discussão sobre os testes de usabilidade com as heurísticas de Nielsen conduzem a uma conclusão aceitável sobre a experiência do utilizador e usabilidade da aplicação.

Quanto à performance da aplicação, esta foi melhorada substancialmente com a ajuda do Google Insight. Os testes de carga/performance (§6.5) demonstraram que a aplicação é bastante rápida, sendo carregada em média em menos de 2 segundos, o que é aceitável. Porque em geral temos menos de 5 segundos do tempo de um visitante da nossa aplicação Web e 74% dos utilizadores vão abandonar se um site demora um tempo de mais de 4 segundos a carregar [26].

7. CONCLUSÕES E TRABALHO FUTURO

Foi com muito gosto que realizei este trabalho com a supervisão do Doutor António Miguel Cruz, a quem agradeço sinceramente pela sua orientação, o seu conhecimento científico, o seu trabalho e dedicação foram essenciais para a concretização de este relatório. Agradeço também à empresa Mainstream pela oportunidade e disposição em realizar este projeto profissional, e também pelo desejo da empresa de continuar com os meus serviços, tendo-me sido comunicado, pelo gerente Professor Joaquim Calhau, ter apreciado a minha atitude e desempenho como prova de competências nesta área. Já são consideradas fases futuras à volta deste trabalho como aquelas citadas mais abaixo.

Para próximas evoluções deste trabalho, considera-se, mais tarde implementar a língua inglesa na aplicação. Uma possibilidade será duplicar os campos de nome, descrição e observação do produto para um nome com sufixo `_EN` (i.e. `nome_EN`, `descr_EN`, `obs_EN`). Terá que fazer-se o mesmo processo para o comércio que duplicará na sua tabela de base de dados um campo `nome_EN` e outro campo `descr_EN`. De resto modificar as variáveis poderá ser feito num editor de texto como o Notepad++ que permite a modificação global de todas variáveis genericamente em todos os ficheiros (i.e trocar a variável “prato” por “produto” e “restaurante” por “empresa” em todas as ocorrências da aplicação Web).

Uma alternativa será recorrer a ficheiros de recursos por língua (português e inglês) para todas as *labels* da aplicação, e a colocação de um código de língua nas tabelas da BD.

Numa segunda fase a empresa Mainstream considera seriamente o alargamento da plataforma de restauração para qualquer outro comércio, associando uma categoria própria a diferentes áreas do comércio (sendo a restauração apenas uma dessas áreas). O restaurante terá equivalência para uma “empresa” e o prato será considerado um “produto” por analogia. A implementação desta segunda fase é considerada bastante rápida por mim próprio, visto que o trabalho já está todo modelado.

A Figura 35 ilustra o protótipo da futura plataforma, intitulada “Em Redor”.



Figura 35 Protótipo futuro de nova plataforma www.emredor.com

Outras melhorias a efetuar, no futuro, poderão passar pelas seguintes tarefas:

A interface de proximidade geográfica na página principal terá que ser otimizada quando houver muitos restaurantes inseridos a aparecer no mapa (i.e. um número superior a 100 ou 1000). Terão de aparecer unicamente aqueles que estão num raio de, por exemplo, 15 ou 20 km do cliente (já está no código uma função para fazer o cálculo com a formula de Haversine); ou então substituir por um marcador de dimensão muito reduzido, como já está preparado no código fonte, para cada localização de restaurante poder aparecer confortavelmente no mapa.

Foi feita a atualização dos ficheiros CSS Bootstrap da versão 3.0.1 para 3.1.1 e será sempre preferível atualizar estes ficheiros otimizados quando aparecem novas versões de Bootstrap. É necessário depois de atualizar os ficheiros CSS do Bootstrap voltar a fazer alguns testes para ver se a interface não é comprometida e fazer as modificações necessárias se a interface for alterada, como por exemplo uma cor de texto não visível. Pondera-se também tentar atualizar alguns ficheiros Javascript.

Foi realizado o trabalho de ter a possibilidade de usar um *Clean URL* como por exemplo www.portal-pt.com/restaurante/7 no lugar de um URL como www.portal-pt.com/restaurante/index.php?num=7 isto usando no diretório `/restaurante/` um ficheiro de `.htaccess` com o código seguinte:

```
Options +FollowSymLinks

RewriteEngine On

RewriteBase /sem3/pratos_em_cartaz/restaurante/

RewriteRule ^([a-zA-Z0-9]+)$ index.php?num=$1
```

Pondera-se trabalhar além deste *Clean URL* um URL mais elaborado como o nome do restaurante no URL. Para chegar a este resultado poderemos por exemplo implementar mais um campo “urlid” único na tabela restaurante com o nome do restaurante escrito lá com a ajuda de uma função que substitui espaços por hífen e caracteres especiais por caracteres simples. Será possível usar um script SQL e a função criada, isto posteriormente ao lançamento da plataforma para executar esta melhoria.

Considerar alguma informação de ajuda se assim for sugerido pelos clientes no momento do lançamento da plataforma. Por exemplo colocar no menu do top um ícone com um “i” (existe no Bootstrap) para abrir uma janela com algum texto de ajuda se acharmos útil.

O *layout* poderá ser remodelado simplesmente, colocando a pesquisa no topo.

Outras alterações aparecerão certamente enquanto a plataforma for lançada.

REFERÊNCIAS

- [1] Pete Cashmore. Dec 11, 2012. "Why 2013 Is the Year of Responsive Web Design" URL:<http://mashable.com/2012/12/11/responsive-web-design> - Link visitado em 10/03/2014
- [2] Visionmobile Ltd. Feb 2014. "Developer Economics Q1 2014: State of the Developer Nation ...the de-facto research series tracking the trends in the app economy". URL:<http://www.visionmobile.com/product/developer-economics-q1-2014-state-developer-nation/> - Link visitado em 10/03/2014
- [3] João Pedro Pereira 10/03/2014. "Venda de tablets em Portugal cresceu 91%" URL:<http://www.publico.pt/tecnologia/noticia/venda-de-tablets-em-portugal-cresceu-91-1627782> - Link visitado em 11/03/2014
- [4] Kristen Purcell et al. March 14, 2011. "How mobile devices are changing community information environments". URL:<http://www.pewinternet.org/Reports/2011/Local-mobile-news.aspx> - Link visitado em 10/03/2014.
- [5] Luke Wroblewski. Oct 18, 2011. "Mobile First" Book. Collection A Book Apart. ISBN: 978-1-937557-02-7
- [6] Jim Cowart. July 2013. "When to Go Native, Mobile Web or Cross-Platform/Hybrid". URL:<http://tech.pro/blog/1355/when-to-go-native-mobile-web-or-cross-platformhybrid> - Link visitado a 10/03/2014
- [7] Visionmobile Ltd. Report Feb 2012. URL:http://www.visionmobile.com/wp-content/uploads/downloads/2013/08/VisionMobile_Cross-Platform_Developer_Tools_2012_v01.pdf - Link visitado a 10/03/2014
- [8] Eva Harb, Paul Kapellari, Steven Luong, Norbert Spot. Responsive Web Design - Version of 06 Dec 2011.
- [9] Jordan Larkin. November 13, 2013. "Do I Need Responsive Web Design for My Website?". URL:<http://EzineArticles.com/8114631> - Link visitado a 11/03/2014
- [10] Marcotte, Ethan. 2011. "Responsive web design". Collection A book Apart (edited by Brown) - New York. URL:<http://www.theusers.info/dev/mobile/RESPONSIVE-WEB-DESIGN-Ethan-Marcotte.pdf>
- [11] De Graeve, Katrien. 2011. "Responsive Web Design." MSDN Magazine. URL:<http://msdn.microsoft.com/en-us/magazine/hh653584.aspx>
- [12] Rivoal, Florian. June 19, 2012. "Media Queries." W3C Recommendation. URL: <http://www.w3.org/TR/css3-mediaqueries>
- [13] Gustavo Neves. 28 Jan 2013. "O que são Media Queries e como elas influenciam o Design Responsivo?". URL: <http://masterstudioweb.com/blog/tag/css/>
- [14] The jQuery Foundation. 2014. URL:<http://jquery.com> - Link visitado a 10/03/2014
- [15] Definição wikipedia. URL: [http://pt.wikipedia.org/wiki/AJAX_\(programação\)](http://pt.wikipedia.org/wiki/AJAX_(programação)) - Link visitado a 10/03/2014
- [16] Definição Wikipedia.URL: http://pt.wikipedia.org/wiki/Document_object_model - Link visitado a 10/03/2014
- [17] Manual oficial do PHP - Prefácio (em português). php.net. URL:http://www.php.net/manual/pt_BR/preface.php - Link visitado em 22/02/2014
- [18] Definição Wikipedia.URL:http://pt.wikipedia.org/wiki/Banco_de_dados - Link visitado a 19/02/2014

- [19] Definição Wikipedia.URL:http://pt.wikipedia.org/wiki/Mapeamento_objeto-relacional - Link visitado a 8/02/2014
- [20] Marcotte, Ethan. May 25, 2010. "Responsive Web Design." A List Apart Magazine – URL:<http://alistapart.com/article/responsive-web-design> - Link visitado em 16/03/2014
- [21] Dennis Altermann. 20/11/2012. Atualizado: 17/11/13. "Design Responsivo". URL:<http://www.midiatismo.com.br/o-mobile/design-responsivo-entenda-o-que-e-a-tecnica-e-como-ela-funciona> - Link visitado a 22/01/2014
- [22] Google Webmaster, Last updated February 11, 2014. URL:<https://developers.google.com/webmasters/smartphone-sites/details#rwd> – Link visitado em 22/01/2013
- [23] Definição Wikipedia.URL: <http://pt.wikipedia.org/wiki/MySQL> - Link visitado a 17/02/2014
- [24] Anne van Kesteren; Simon Pieters ; W3C. (25 de maio de 2010). «HTML5 differences from HTML4». URL:<http://www.w3.org/TR/2011/WD-html5-diff-20110525/> - Página visitada em 26/02/2014.
- [25] Nielsen, J. (1994b). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, NY. (extracto no URL:<http://www.nngroup.com/articles/ten-usability-heuristics/>)
- [26] Luke Wroblewski. September 24, 2012. "Breaking Development: Responsive Design Vs Separate Mobile Sites". URL: <http://www.lukew.com/ff/entry.asp?1631> - Link visitado a 5/03/2014